

**INSTRUMENTATION FAULT SIMULATION  
TRAINING MODULE (FAST.tool)**

By

**NURUL ATIKAH BINTI MAHZAN**

**DISSERTATION REPORT**

**Submitted to the Electrical & Electronics Engineering Programme  
in Partial Fulfillment of the Requirements  
for the Degree  
Bachelor of Engineering (Hons)  
(Electrical & Electronics Engineering)**

**JUNE 2008**

**Universiti Teknologi Petronas  
Bandar Seri Iskandar  
31750 Tronoh  
Perak Darul Ridzuan**

**© Copyright 2008**

**by**

**Nurul Atikah Bt Mahzan, 2008**

# **CERTIFICATION OF APPROVAL**


## **INSTRUMENTATION FAULT SIMULATION TRAINING MODULE (FAST.tool)**

by

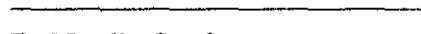
Nurul 'Atikah Bt Mahzan

A project dissertation submitted to the  
Electrical & Electronics Engineering Programme  
Universiti Teknologi PETRONAS  
in partial fulfillment of the requirement for the  
Bachelor of Engineering (Hons)  
(Electrical & Electronics Engineering)

Approved:



Dr Muhammad Asif Sadiq  
Project Supervisor



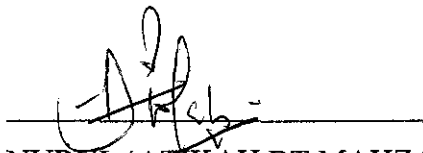
Dr Nordin Saad  
Project Co-Supervisor

UNIVERSITI TEKNOLOGI PETRONAS  
TRONOH, PERAK

June 2008

## **CERTIFICATION OF ORIGINALITY**

This is to certify that I am responsible for the work submitted in this project, that the original work is my own except as specified in the references and acknowledgements, and that the original work contained herein have not been undertaken or done by unspecified sources or persons.



NURUL 'ATIKAH BT MAHZAN

## **ABSTRACT**

‘Instrumentation Fault Simulation Training Module’ or known as FAST.tool is essentially a learning tool which serves as a training kit for instrument troubleshooting. This tool is enthused by numerous complaints concerning new engineers who are inadequately performing troubleshooting. Believing that the engineers’ deficiencies are due to lack of experience and substantial training, this tool will optimistically solve the problem. Understanding of various type of instrumentation faults are the key points in order to accomplish this project as well as good knowledge in Visual Basic scripting and coding. To accomplish the project, the prototyping process model approached is applied. The model process begins with a requirements capture activity, followed by a quick design and build of a prototype or mock-up of the product. After analyzing the first built model, further modifications to the requirements are generated and the process begins again to build the second model. These processes continue until the final model satisfied the objectives of the project. Taking into consideration all interactive enhancement suggestions towards the project, new and better user interface design is created and introduced. This project aims to lay the solid foundation for the troubleshooting skill development using multimedia aid become to overcome the troubleshooting deficiency.

## **ACKNOWLEDGEMENTS**

I would like to take this opportunity to acknowledge and thank everyone that has given me all the supports and guidance throughout the period of completing the final year project. Firstly, many thanks to the university and the Final Year Project coordinators that have coordinated and made the necessary arrangements for me to complete this project.

I must also acknowledge the endless help and support received from my supervisor, Dr. Muhammad Asif Sadiq and co-supervisor, Dr. Nordin Saad throughout the whole period of completing the final year project. Their guidance and advices are very much appreciated. Apart from that, many thanks to my former supervisor at Shell Refining Company Berhad, Port Dickson, Dr. Khairil Osman for giving me the inspiration and idea starting from the very beginning of the project until the compilation stage. Apart from Dr. Khairil, I must also thank the engineer at Shell Refining Company, Ms. Farrah Diva Herman who helped me to get information on instrumentation and Mr. Redzuan Mansor who gave a lot of feedback for this project.

Last but not least, I would like to thank my family and friends for their support and motivation, which pushed me on to complete this project.

## **TABLE OF CONTENTS**

<b>ABSTRACT .....</b>	<b>iv</b>
<b>LIST OF TABLES .....</b>	<b>viii</b>
<b>LIST OF FIGURES .....</b>	<b>ix</b>
<b>LIST OF ABBREVIATIONS.....</b>	<b>xi</b>
<b>CHAPTER 1 INTRODUCTION .....</b>	<b>1</b>
1.1 Background of Study .....	1
1.2 Problem Statement.....	1
1.3 Objectives and Scope of Duty .....	2
1.3.1 Objectives .....	2
1.3.2 Scope of Duty: Design/Software Development.....	3
1.3.3 Scope of Duty: Instrument Troubleshooting.....	3
<b>CHAPTER 2 LITERATURE REVIEW .....</b>	<b>5</b>
2.1 Introduction .....	5
2.2 Issues on Application Software .....	6
2.3 Approaches to Development of Application Software.....	6
2.4 Instrumentation Troubleshooting Practice .....	7
<b>CHAPTER 3 METHODOLOGY AND PROJECT WORK.....</b>	<b>9</b>
3.1 Research on Instrumentation Troubleshooting.....	10
3.1.1 Common Faults Identification .....	10
3.2 Visual Basic Skill Development.....	12
3.2.1 Information Analysis .....	13
3.2.2 Design .....	14
3.2.3 Multimedia Authoring .....	14
3.3 Integrate with database .....	15

**CHAPTER 4 RESULTS AND DISCUSSION .....18**

4.1 Previous Results: Final Year Project I..... 18

4.2 Current Results and Discussion: Final Year Project II..... 21

**CHAPTER 5 CONCLUSION AND RECOMMENDATION .....28**

5.1 Conclusion ..... 28

5.2 Recommendations for Future Work ..... 28

**REFERENCES .....30**

**APPENDICES .....31**

Appendix A Gantt chart..... 32

Appendix B MyProgram version 1.3 interface example .....36

Appendix C FAST.tool version 1.4 interface and Coding ..... 37

Appendix D Story Boards ..... 48

Appendix E Flowchart..... 51

## **LIST OF TABLES**

Table 1 : Table of current industrial market conditions .....	2
Table 2 : Problem - Transmitter read zero volts.....	12
Table 3 : Table of Junction Box Cabinet Values.....	17
Table 4 : Description for main frame .....	48
Table 5 : Description for junction box cabinet.....	50



## LIST OF FIGURES

Figure 1 : Flowchart of project methodology.....	10
Figure 2 : A digital multimeter.....	11
Figure 3 : Example of a Loop Diagram; Loop 23FI-207 .....	11
Figure 4 : Prototyping Process Model.....	13
Figure 5 : Example of 4-20 mA Loop Diagram .....	16
Figure 6 : Example of a Loop Diagram; Loop 72 FI-195 .....	16
Figure 7 : MyProgram Version 1.0; 24 Volts.....	18
Figure 8 : MyProgram Version 1.0; 0 Volt .....	19
Figure 9 : MyProgram Version 1.3; Main Form .....	20
Figure 10 : MyProgram Version 1.3; choose a level.....	20
Figure 11 : FAST.tool Version 1.4; Splash Screen .....	21
Figure 12 : FAST.tool Version 1.4; Login Form .....	21
Figure 13 : FAST.tool Version 1.4; Welcome Screen .....	22
Figure 14 : FAST.tool Version 1.4; 'Loop Drawing' button .....	23
Figure 15 : Example of loop drawing; Loop 23-FI-207.....	23
Figure 16 : Choosing the correct transmitter based on loop drawing .....	24
Figure 17 : Click on the center of the rectangular Junction Box block.....	24
Figure 18 : FAST.tool Version 1.4; Default Junction Box Cabinet Form .....	25
Figure 19 : FAST.tool version 1.4; Checking the right transmitter .....	25
Figure 20 : FAST.tool version 1.4; getting the DMM reading .....	26
Figure 21 : Example of database; data for loop 23FI-207.....	27
Figure 22 : MyProgram Version 1.3; problem solved.....	36
Figure 23 : FAST.tool Version 1.4; Junction Box Cabinet Form .....	40
Figure 24 : Main Frame.....	48

## LIST OF ABBREVIATIONS

DCS	Distributed Control System
DMM	Digital Multi Meter
FTA	Field Terminal Assembly
GUI	Graphical User Interface
I/O	Input/output
IC	Integrated Circuit
JB	Junction Box
MDF	Main Distribution Frame
P&ID	Piping and Instrumentation Diagram
PLC	Programmable Logic Control
VB	Visual Basic

# **CHAPTER 1**

## **INTRODUCTION**

### **1.1 Background of Study**

Learning process these days is more convenient with the use of computer-based instruction. With multimedia aid, the learning process becomes easier, interesting and efficient. Instead of text presentation, multimedia programs could use pictures, sounds or even video to further explain about the subject.

Employing the multimedia aid into the instrument troubleshooting training is seen to be a beneficial practice. In instrumentation, '*troubleshooting*' is defined as the action taken to fix any failure occurs while '*failure*' is described as the condition of not achieving a desired state or function. Dealing with failures is a troubleshooter's business, and to troubleshoot successfully, he/she must first understand how failure occurs. Specifically in a process plant, failures can occur due to factors such as a faulty component (hardware), an incorrect line of programming code (software), or a human error (systematic). A system can even have a functional failure when it is working properly but is asked to do something it was not designed to do or when it is exposed to a transient condition that causes a momentary failure.

### **1.2 Problem Statement**

Instrument engineers play an important role towards plant safety as well as its productivity. One vital task as an instrument engineer is to troubleshoot

whenever there is unplanned event or failure of instrument device happened in plant. However, there are numerous reports from global companies complaining that new engineers perform defectively while troubleshooting in plant. They are proven to have convincing fundamental of knowledge but lack of troubleshooting skills. Table 1 below summarizes the reports.

Table 1 : Table of current industrial market conditions

STAFF	CONSEQUENCE
<ul style="list-style-type: none"> <li>• Shortages of qualified staff resources</li> </ul>	<ul style="list-style-type: none"> <li>• Imposes bigger emphasis on both initial and continuous training</li> </ul>
<ul style="list-style-type: none"> <li>• Recruitment of low quality and non-experienced or with little experience personnel</li> <li>• Problems in recruiting supervisory and middle management</li> <li>• Shortage in welding and instrument supervisors</li> </ul>	<ul style="list-style-type: none"> <li>• Mobilize early to allow for longer training curve to perform as expected</li> <li>• Need better financial and other incentives to recruit and retain employees</li> </ul>

Therefore, this tool would be targeted at new engineers and technicians, as well as for refresher for the experienced personnel.

### 1.3 Objectives and Scope of Duty

#### 1.3.1 Objectives

The aim of the project is to develop a training tool for instrument troubleshooting. This tool can be used to train the user on how to perform troubleshooting based on real situation. The main features include:

- i. Able to randomly generate faults in a single/multiple loops and require the user to troubleshoot and locate these faults.
- ii. Able to generate different types of instrument faults based on usual cases as to expose user to real troubleshooting situation.

- iii. Exposes user to common instrument terminologies such as loop drawings, Distributed Control Systems (DCS), Field Terminal Assembly (FTA), Junction Box (JB) and etc.
- iv. Integrated with adjustable database for administrator convenience.
- v. Overall performance can be saved or printed for future reference.

### ***1.3.2 Scope of Duty: Design/Software Development***

Meanwhile, the scope of duty is to focus on both mastering the instrumentation troubleshooting case and developing software that is using multimedia element. The training module is designed in such a way to inquire the user to locate the faults and solve the problems.

This project must be feasible to be completed within duration of four months. Further description for the time detail is as indicated in Gantt chart in Appendix A.

### ***1.3.3 Scope of Duty: Instrument Troubleshooting***

The very fundamental foundation in instrumentation field is to get the grasp of the relevant process variables, then to look at their measurements and the essentials of the systems that control them. Troubleshooting basics cover the systematic approach to information gathering, fault diagnosis and decision-making.

Instrument troubleshooting is emphasizing on gathering relevant information and using it to prove where the fault isn't, thereby eliminating false decisions and 'red herrings'. For implementing the right solution, the engineer has to learn from the experience and look on how to prevent a recurrence.

Taking the case of an apparent Programmable Logic Control (PLC) fault, first level trouble-shooting can eliminate the PLC from the actual fault condition

and pin-point the section of a plant where the real fault may be located; typically a range of 5 to 10 components. However PLC is acknowledged as a robust system which seldom falls into failure.

In the training module later, the advanced section will focus more on tracing faults to the final component and might require using more sophisticated equipment, and/or debugging. The point is that if the PLC programming was correctly commissioned, it will not be the cause of the problem.

Similarly, first line motor faults can be related to individual motors, drive circuitry, relays and switches etc. The advanced troubleshooting will cover tracing the fault to the specific integrated circuit (IC) or other drive component, such as a thyristor or fuse. In some cases, a recalibration of the drive might also be required. The assumption here is that most faults can be traced to fuses, misaligned components and loose connections etc. If the fault goes beyond this and is caused by a faulty component, it needs to be verified by a person with more expertise. It is rare to have incorrect configuration or malfunction coding to be the actual problem.

After all a good instrument engineer should be able to analyze any problems and suggest a right solution quickly to prevent lose and longer plant down-time. He/she is expected to accurately interpret drawings and documentation, reliably explain the nature of faults, how they should be repaired and how the system should be tested on completion and competently diagnose and fix fault conditions. Further he/she should be able to demonstrate a working knowledge of how instrumentation, electrical and process control systems are interfaced and eventually compile concise faults analysis reports and make recommendations to prevent a recurrence.

## **CHAPTER 2**

### **LITERATURE REVIEW**

#### **2.1 Introduction**

Troubleshooting in instrumentation covers a lot of things including the fundamental electrical and electronic engineering knowledge. If the troubleshooting was to put as on-the-job training, the engineer will learn to explain how voltage, current and resistance are related by Ohm's Law and apply these principles to the design and function of electronic circuits, use electrical measuring instruments for troubleshooting, work with electronic measurement components such as multimeters, oscilloscopes, bridge-type instruments and digital test equipment and as well explain the operation of electronic equipment and systems such as servo and control systems, numerical control systems, computers, programmable controllers, and microprocessors used in industry.

The concept of troubleshooting can be taught and learned. It has about similar basic starting which is looking into the loop drawing to understand the process line and further actions are diverge with respect to different instrument variables such as flow, level, temperature or pressure and the technologies applied in the plant.

These days, there are a lot of ways to deliver a subject. Troubleshooting can be a subject to be delivered to the new engineers. A typical example is learning in classroom. This method is widely used over the world to teach a student about a subject. But how can we educate someone on skill specifically troubleshooting skill? The teacher-student approached does not seem efficient when it comes to educate someone with good knowledge but lack of skill. The better approached will

be a multimedia classroom and let the graphical things do all the teachings.

## **2.2 Issues on Application Software**

According to England and Finney (1999), “graphics based navigation has several advantages over text based” which include taking up less space, not language specific, more visual and appealing as well as making access through an application (in this case a software) much easier. This is why graphical approach is used in this project rather than textual. User will easily understand the program and catching up with it faster.

Meanwhile, according to Gagne and Briggs (1985), learning is facilitated by a specific order of instructional methods and events of instructions. Basically there are nine ordered events of instruction as classified by both of them. All of these factors must be taken care into consideration in creating multimedia learning software. The factors are listed in methodology later in this report.

Constantine (1995) points out that the reality is that a good user interface allows people who understand the problem domain to work with the application without having to read the receive training. A fundamental reality of application development is that user interface is the system to the users. What users want is for “developers to build applications that meet their needs and that are easy to use” (Buis (1996)). Too many developers think that they are artistic geniuses, as they do not bother to follow user interface design standards or invest the effort to make their application usable, instead they mistakenly believe that the important thing is to make the code clever or to use a really interesting color scheme. For most people the user interface is the application software.

## **2.3 Approaches to Development of Application Software**

Ambler (2000) has proposed several techniques for a good user interface design. It is important to ensure that the software user interface works consistently.



For example double-clicking on items in one list and have something happen then the user should be able to double-click on items in any other list and have the same sort of thing happen. Buttons are put in consistent places on all of the windows, use the same wordings in labels and messages, and use a consistent color scheme throughout. Consistency in the software's user interface allows the users to build an accurate mental model of the way that it works, and accurate mental models lead to lower training and support costs. Tognazzini (1991) and Laurel (1991) supported the statements.

Frequently a complex software system can be understood more easily if the user interface is depicted in way that resembles some commonplace system (Talin, 1998). Taking this into consideration, the software is presented using block diagrams, standard P&ID diagrams and standard figure of instrumentation equipments and tools.

## **2.4 Instrumentation Troubleshooting Practice**

Mostia (2000) stated that regarding the troubleshooting deficiency, some people argue that troubleshooting is an art while in fact successful troubleshooting depends more on logic and knowledge. Because of this, troubleshooting can be taught and developed. In general, the troubleshooting expertise can be achieved by experiences, apprenticeships, mentoring, classroom instruction and also individual study.

A lot of trainings are provided for engineers and technician to improve their troubleshooting skill. ISA – The Instrumentation, Systems, and Automation Society is a body which provides programs like ISA's Certified Control System Technician (CCST) tests reward training at home, on the job and in classrooms. Through these programs, engineers and technicians can develop their skills in troubleshooting and perform better job at workplace. IDC Technologies, an Internationally Accredited Professional Training Organization also offers interactive workshop which uses a systems approach to troubleshooting. The workshop is designed to encourage

participants to take a new look at the methodology of faultfinding and rectification on their plant. Beginning with covering the types of equipment, they then focus on the first line troubleshooting, then the advanced level and finally work through some typical examples.

Therefore by implementing the aid of multimedia to the instrumentation troubleshooting is a good method as the new engineer is expected to know only a few things in instrumentation field. After familiarizing himself with the module later, then he is prepared to undergo any workshop or training to improvise his skill.

**CHAPTER 3**  
**METHODOLOGY AND PROJECT WORK**

The methodology of this project in general is summarized in the flowchart below.

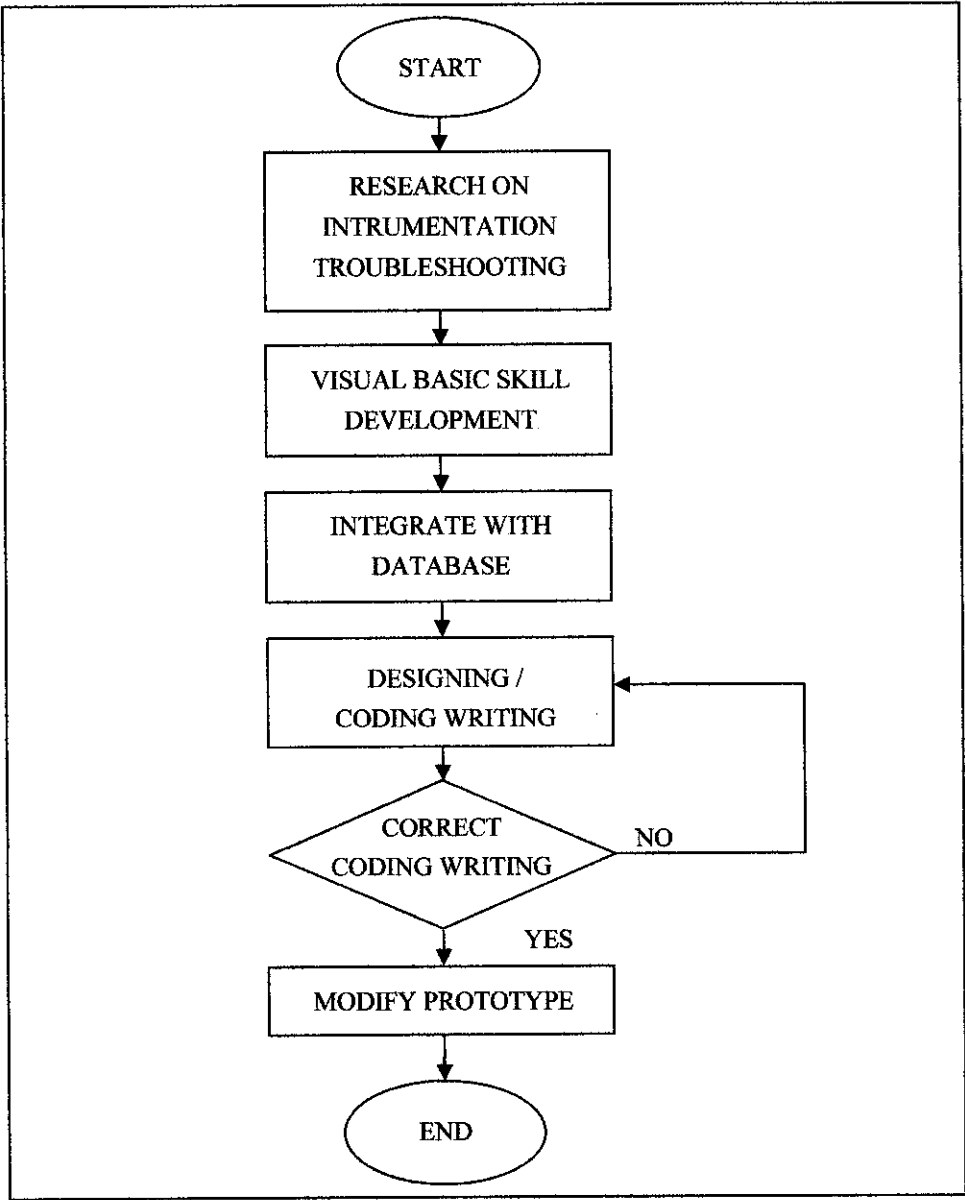


Figure 1 : Flowchart of project methodology

### **3.1 Research on Instrumentation Troubleshooting**

The research project is focusing on the Instrumentation part. The research activities include referring to books, journals, magazines and also Instrument CDs by Clarity Multimedia, Coastal Training Technologies Corp. ® 2003.

The initial part was to configure the instrument loop diagrams. There were a lot of instrument loop examples to be listed but for the purpose of this project, a few of simple loop diagrams are sufficient to achieve the objectives. These loop diagrams can generate either single or multiple faults within it. This training module is designed based on case histories of troubleshooting methods and problem solving techniques used by real instrument technicians in their daily work.

#### **3.1.1 Common Faults Identification**

Although instrumentation systems are all different, the process of troubleshooting them follows the traditional path, i.e. start at the source of the signal and work through the signal path to the end. In order to do this an engineer/technician would need a wiring diagram and a digital multimeter.

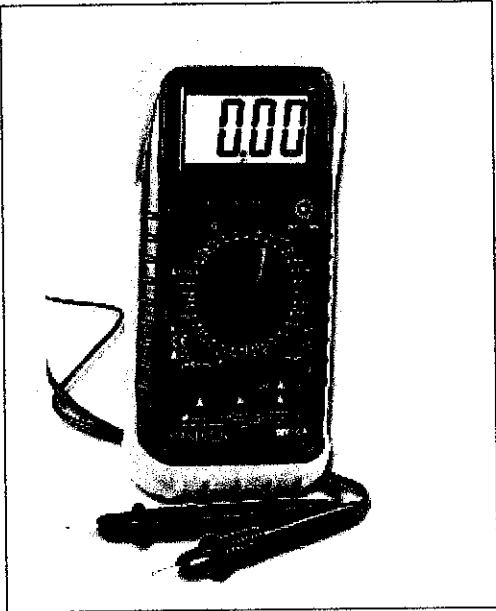


Figure 2 : A digital multimeter

Based on one example of loop diagram (refer figure 3), there are a few possible faults can be found in loop 23-FI-207, and table 2 shows the steps and actions taken to troubleshoot the problem and the result sequenced.

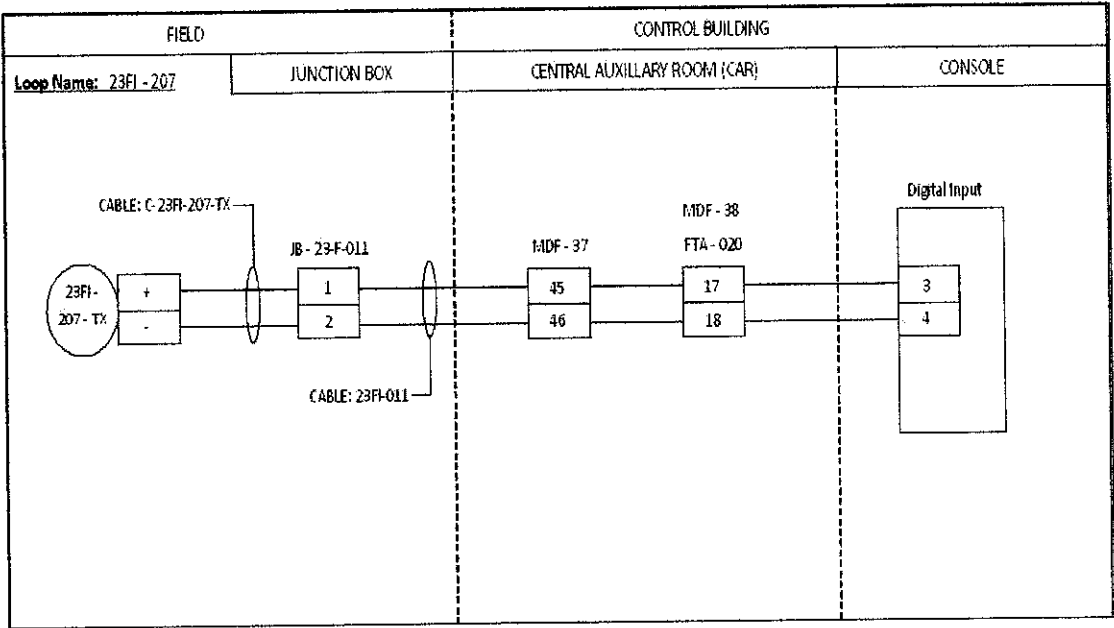


Figure 3 : Example of a Loop Diagram; Loop 23FI-207

Table 2 : Problem - Transmitter read zero volts.

ACTION		RESULT
1	Examine transmitter field output to verify output.	Zero output verified – no voltage present at transmitter, possibly due to loss of power.
2	Review loop drawing to find loop fuse location, then check fuse.	Fuse status indicator shows a blown fuse (note that the fuse is not shown in figure 3).
3	Replace fuse (1 at JB 23-F-011).	Blows again.
4	Examine transmitter. (NOTE: Start at one end where short is most likely to happen, i.e. in this case is in the field).	No indication of short to ground.
5	Examine field indicator located at control valve (note that the control valve is not shown in figure 3).	Cover loose – water and corrosion in indicator are causing short to ground.
6	Replace local indicator and install new fuse.	Problem solved.

This loop 23- FI-207 example was also used in the first attempts in proving the concept.

### 3.2 Visual Basic Skill Development

The designing stage project would revolve around the prototyping process model; an appropriate model when the requirements are incompletely specified. As for this software will be further refine after the prototype has been analyzed. The prototyping process model is shown below.

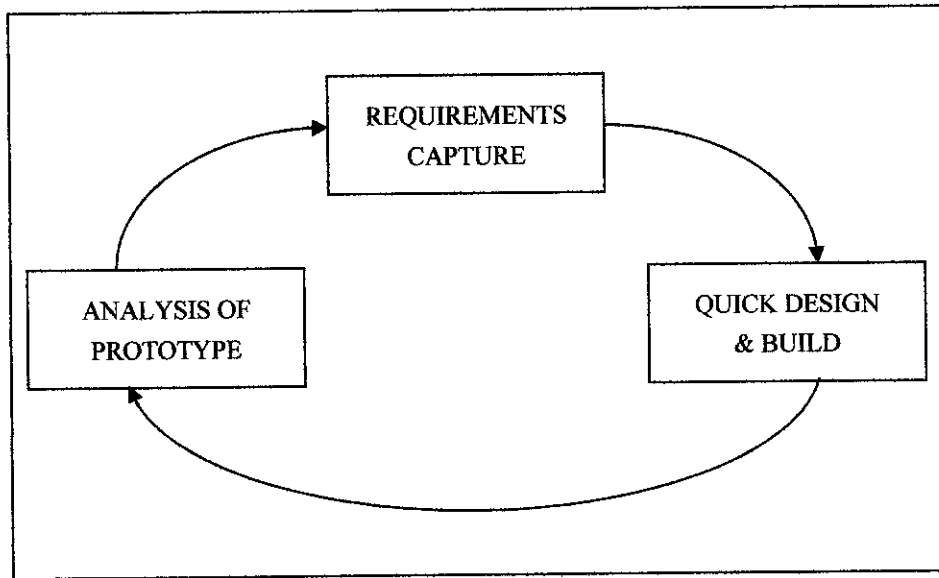


Figure 4 : Prototyping Process Model

The prototyping model encompasses the following activities:

### 3.2.1 *Information Analysis*

The instrumentation part was classified into four major phases previously. The four phases in sequence order are; to identify various loop configurations, to understand the possible faults in different difficulty levels for different loops, to identify the common problems in plant and to list down the troubleshooting algorithm in a flowchart form. After gathering more information and doing some research and review, the scope was narrowed down to identification of different types of instrument faults and list down the troubleshooting algorithm only. These two phases basically covered all requirements needed to perform troubleshooting.

The multimedia information gathering process is conducted to understand the nature of the software to be built, the problem statement. The solutions taken must comply with the objectives of the research project.

### 3.2.2 Design

From the literature review, Gagne and Briggs 9 Instructional Events would be used. The steps of events involved in this model are:

- i. Gaining attention (reception)
- ii. Informing learners of the objective (expectancy)
- iii. Stimulating recall of prior learning (retrieval)
- iv. Presenting the stimulus (selective perception)
- v. Providing learning guidance (semantic encoding)
- vi. Eliciting performance (responding)
- vii. Providing feedback (reinforcement)
- viii. Assessing performance (retrieval)
- ix. Enhancing retention and transfer (generalization)

These nine events are used to incorporate the necessary conditions for learning into the training module as well as abiding with the basis for designing instruction. Subsequently, based on these nine events, story boards (see Appendix D) together with the flow of courseware are produced, with the implementation of User Interface Design (UID) principles cited from literature review.

### 3.2.3 Multimedia Authoring

The software that would be used to produce the training module comprises of the following:

- i. Microsoft Visual Basic 6.0

Microsoft Visual Basic is a powerful Microsoft tool which employs visual



basic, a high-level programming language. It is not only allows programmers to create simple graphical user interface (GUI) applications, but can also develop complex applications or prototyping as well. Programming in VB is a combination of visually arranging components or controls on a form, specifying attributes and actions of those components, and writing additional lines of code for more functionality. Since default attributes and actions are defined for the components, a simple program can be created without the programmer having to write many lines of code.

ii. Adobe Photoshop 6.0

Adobe Photoshop is used for media editing, animation and authoring. In this project, it was used to create and refine graphics. These graphics are then optimized to make the file size smaller before being imported into Microsoft Visual Basic.

Photoshop provides special feature for non-linear editing and special effects services, such as backgrounds, textures and etc. it can utilize the color models RGB (an additive color model in which red, green and blue light are added together in various ways to reproduce a broad array of colors), CMYK (short for cyan, magenta, yellow and key (black) and often referred to as process color or four color), grayscale, binary bitmap and duotone.

### 3.3 Integrate with database

The first model of this project is to have a preset value and is stored inside the coding. Please refer to figure 5 below for further clarification. Let's take a simple example (read the diagram from left to right), the fixed value (written in coding) at termination 11 & 12 in JB- 10 is 4.093 mA and at termination 22 & 23 is 18.974 mA. However it became difficult when there are too many values and figures that need to be preset and checked everytime the coding is modified. Therefore, having a database is a brilliant idea as it is proven for lesser time and cost consuming.

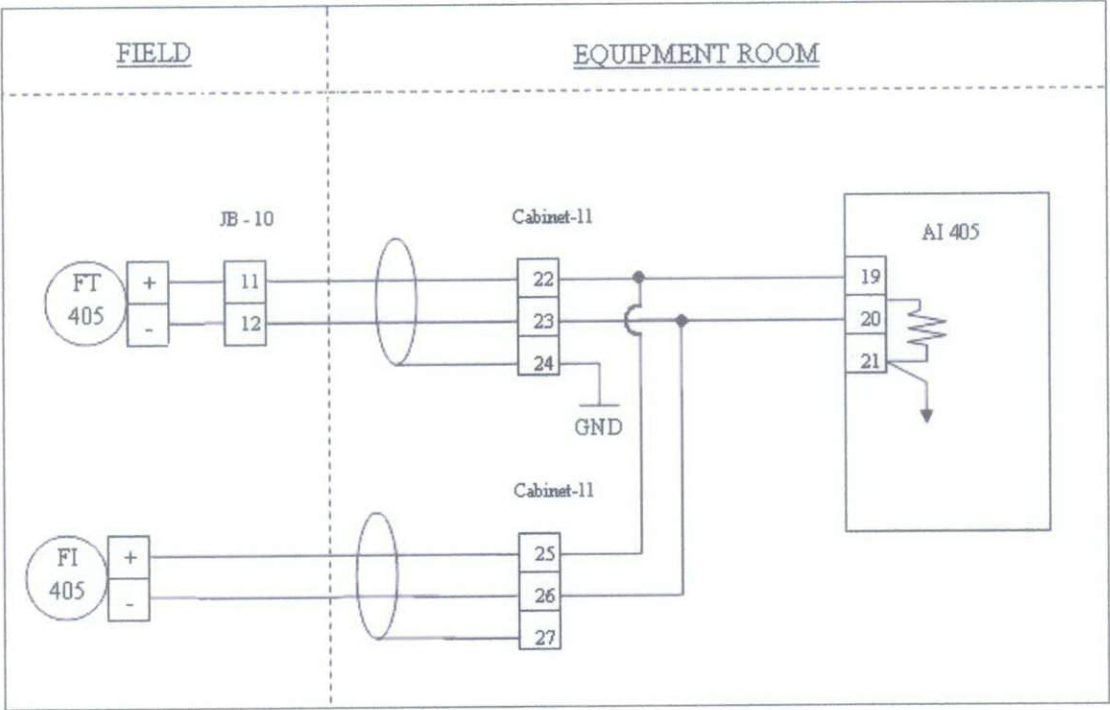


Figure 5 : Example of 4-20 mA Loop Diagram

After going through the model/prototype modification, the design scope is expands to be integrated with an alterable database. This database will organize the collection of values and figures which can be assess and retrieve by the program.

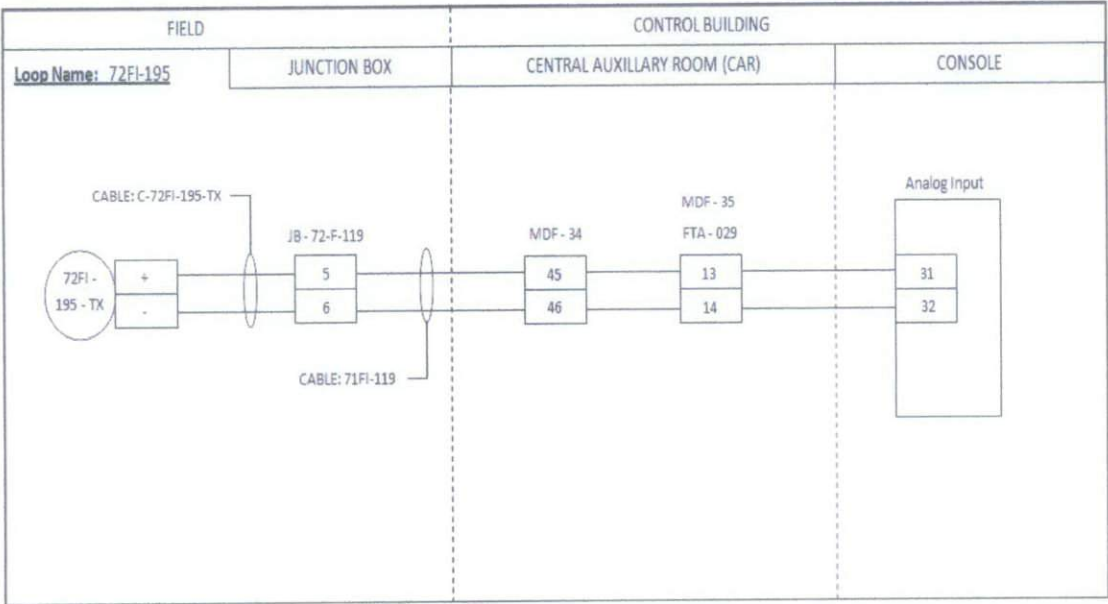


Figure 6 : Example of a Loop Diagram; Loop 72 FI-195

This is how the database and program is linked. For instance, take an example on loop 72 FI-195 in figure 6 above. An instrument engineer is supposed to know how to extract data and interpret it from a loop drawing. In this loop 72 FI-195 case, it reads as (from left to right) Transmitter 72 FI-195 is connected to the JB 72 F-119, connected to MDF-34 and MDF-35 and finally to the DCS. Meanwhile the numbers in the rectangle box represent the termination points that connect the system. For example, the fixed value (preset in database) at termination 5 & 6 in JB 72-F-119 is 18.810 mA and at termination 45 & 46 in MDF-34 is 17.590mA. Table 3 is showing some extracted values from the program's database. The program will retrieve the value for each fault case from this database. The significance of this database is further explained in the result and discussions part.

Table 3 : Table of Junction Box Cabinet Values

2_JBCabinet_Value												
ID	Tag No	AT1-T2	AT3-T4	AT5-T6	AT7-T8	AT9-T10	others	DT1-T2	DT3-T4	DT5-T6	DT7-T8	DT9-T10
2	12TI-154	2.312	19.418	3.506	18.786	18.976	-----	1.245	23.756	0.557	22.986	23.749
3	11TI-100	18.986	19.556	18.810	19.014	0.601	-----	23.512	19.982	23.786	0.019	0.405

The advantage of having a database for this program is the administrator or the supervisor can simply revise the program and modify the value in the database for his/her own purposes.

## CHAPTER 4

### RESULTS AND DISCUSSION

#### 4.1 Previous Results: Final Year Project I

In Final Year Project part 1, the project started with research and findings stage followed with prototype development. It first developed using REALbasic 2007 software which also a visual basic programming software. The first version (refer figure 7 and 8) was constructed to prove the concept of the training module which can arbitrarily generate faults within the loop drawing. It took about 7 days to build the first form.

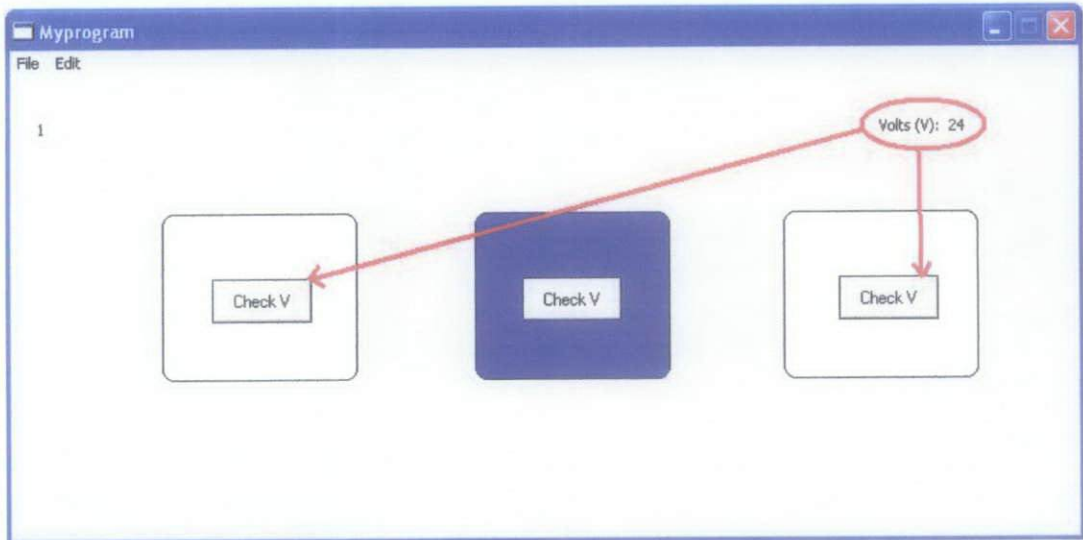


Figure 7 : MyProgram Version 1.0; 24 Volts



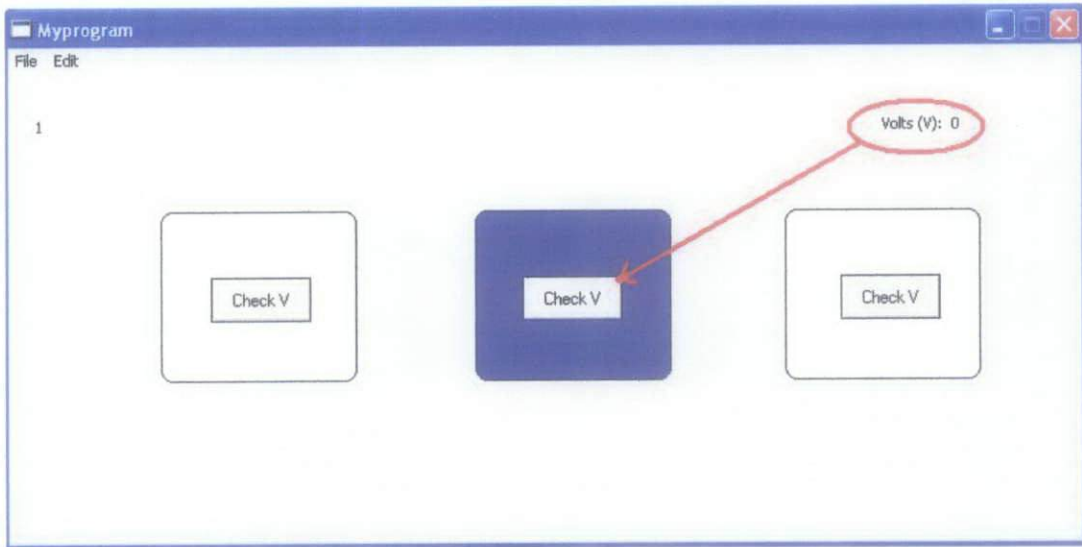


Figure 8 : MyProgram Version 1.0; 0 Volt

The blue rectangle box shows that the error is at that point while the white rectangle boxes show normal readings. Once the program is started, the fault will lie at any point without predetermined.

The second version (refer figure 9) was developed using Microsoft Visual Basic 2005. At this time more forms were inserted and the interactivity of the program with the user were improved. The second version later upgraded to third version which takes about the same feature with second version except for a few upgraded feature. Up to third version, the program was developed with three different difficulty levels (refer figure 10). They are 'Beginner Level', continued by 'Intermediate Level' and finally will be enhanced to 'Expert Level'.

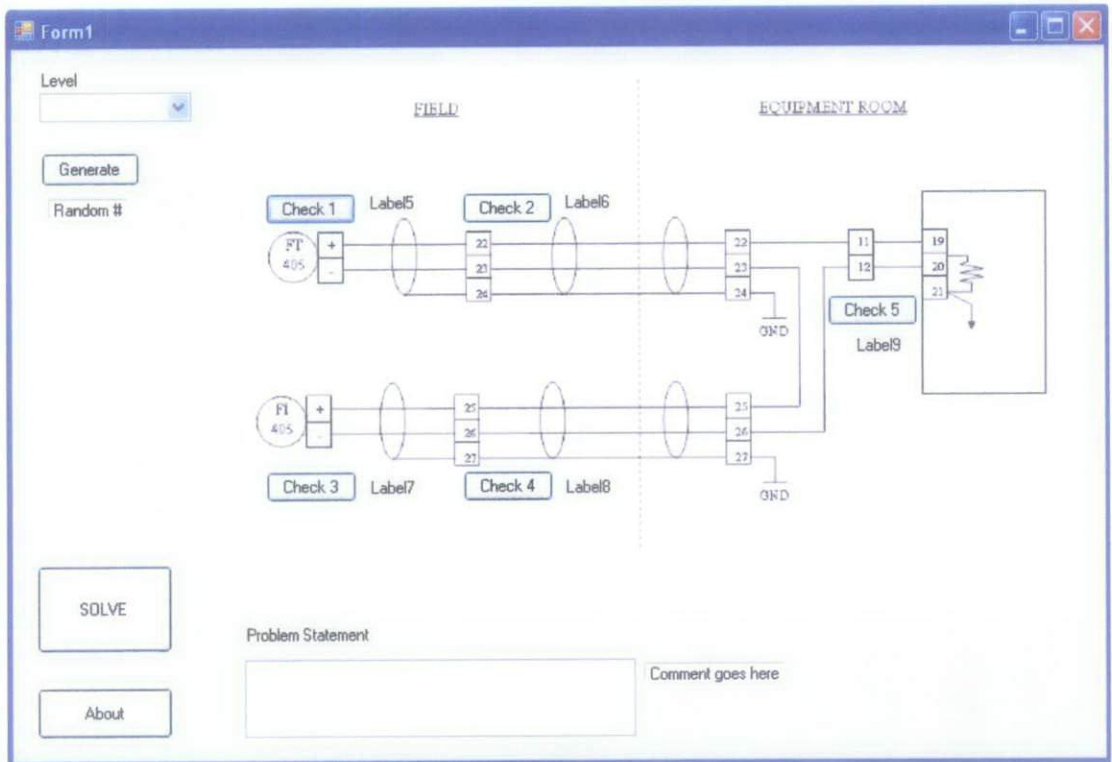


Figure 9 : MyProgram Version 1.3; Main Form

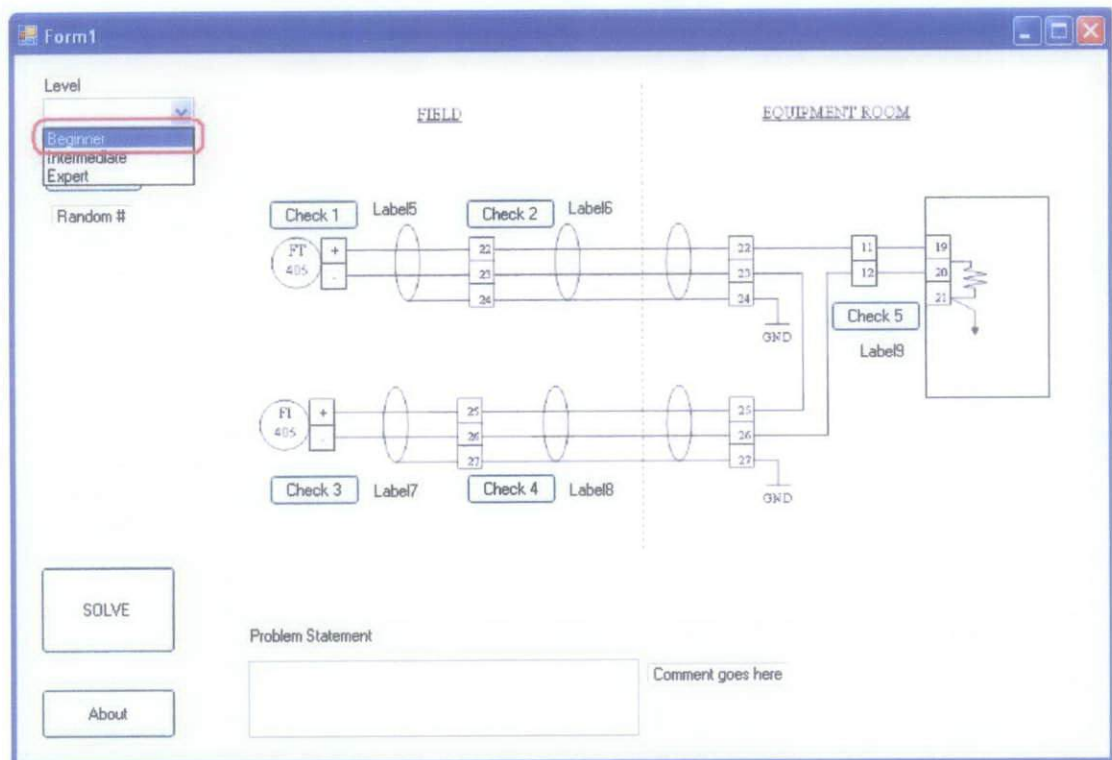


Figure 10 : MyProgram Version 1.3; choose a level

## 4.2 Current Results and Discussion: Final Year Project II

For Final Year Project part 2, a few enhancements have been done to improve the features of the program. In general, the additional features include generating random problems without different difficulty levels, database integrating, auto-saved form and a few more. Follows are the features of the enhanced model explained in detail.

To begin with, the program will prompt the user with a splash screen(refer figure 11) and a login form(refer figure 12). The user has to input his/her name and ID No. for recording purpose as shown in the figure 12.



Figure 11 : FAST.tool Version 1.4; Splash Screen



Figure 12 : FAST.tool Version 1.4; Login Form

Then the user will be prompted with the main form. A welcoming message instructing the user to click on the ‘Problem’ button will appear (refer figure 13). Then the user will proceed as per guidance.

‘Help’ button on the middle left of the form provides instructions and tutorial to guide a first time user performing this exercise. The simple tutorial will assist user to familiarize with the figures, button and caption comprised in the program. This tutorial will facilitate the user with the step-by-step procedure on operating the program.

The current prototype, version 1.4 no longer divided into 3 different levels. It is divided into two levels virtually. The first 3 random problems will be as easy as a guide to familiarize user to the program. Once he/she completed the first 3 problems, the next problem generated will be much tougher. And the program will continue until he/she completed 20 problems as per planned.

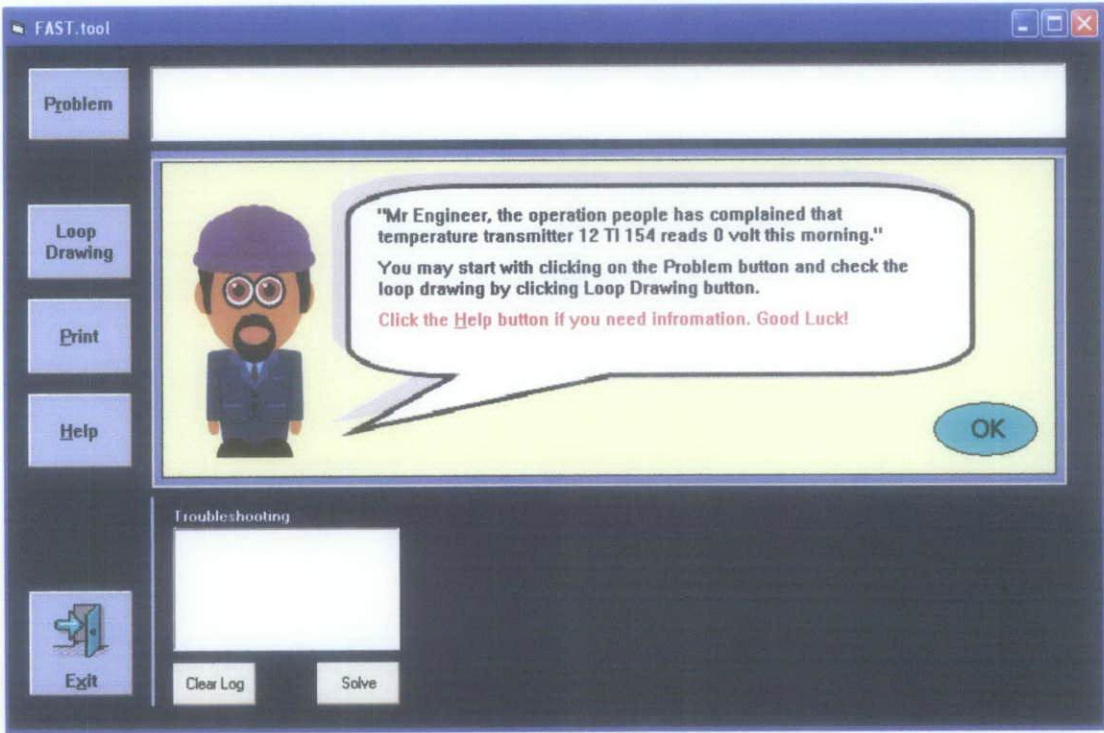


Figure 13 : FAST.tool Version 1.4; Welcome Screen

The user will proceed with checking on the loop drawing by clicking



‘Loop Drawing’ button(refer figure 14). The particular loop drawing will appear as shown in figure 15. Using the loop drawing, the user has to choose for the correct termination point. In this example, the correct transmitter is 23 FI-207 and the correct JB is 23 F-011.

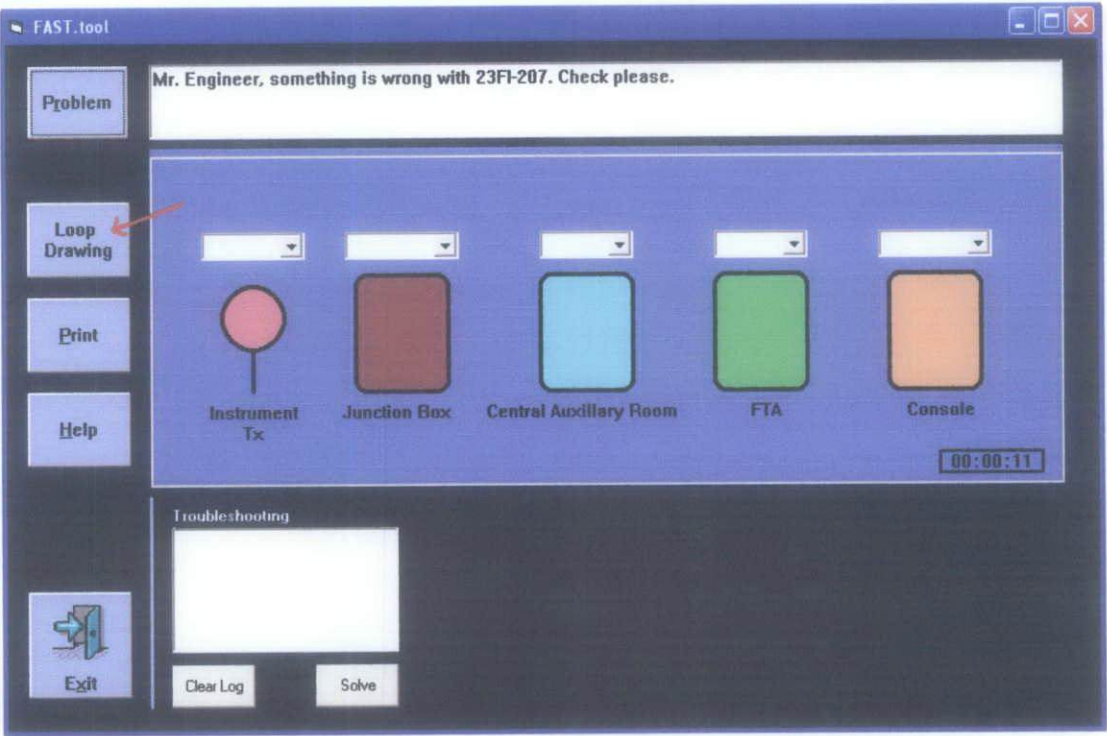


Figure 14 : FAST.tool Version 1.4; ‘Loop Drawing’ button

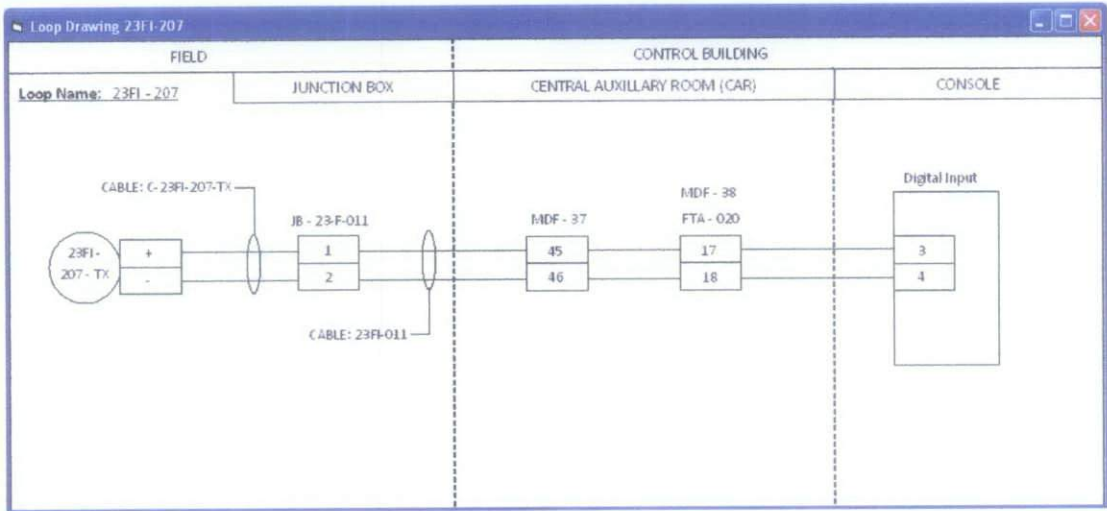


Figure 15 : Example of loop drawing; Loop 23-FI-207

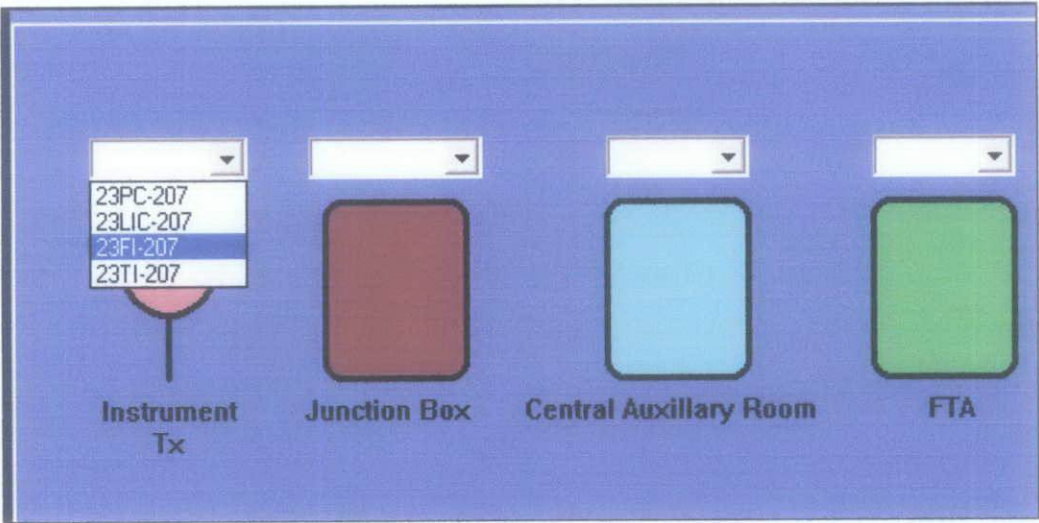


Figure 16 : Choosing the correct transmitter based on loop drawing

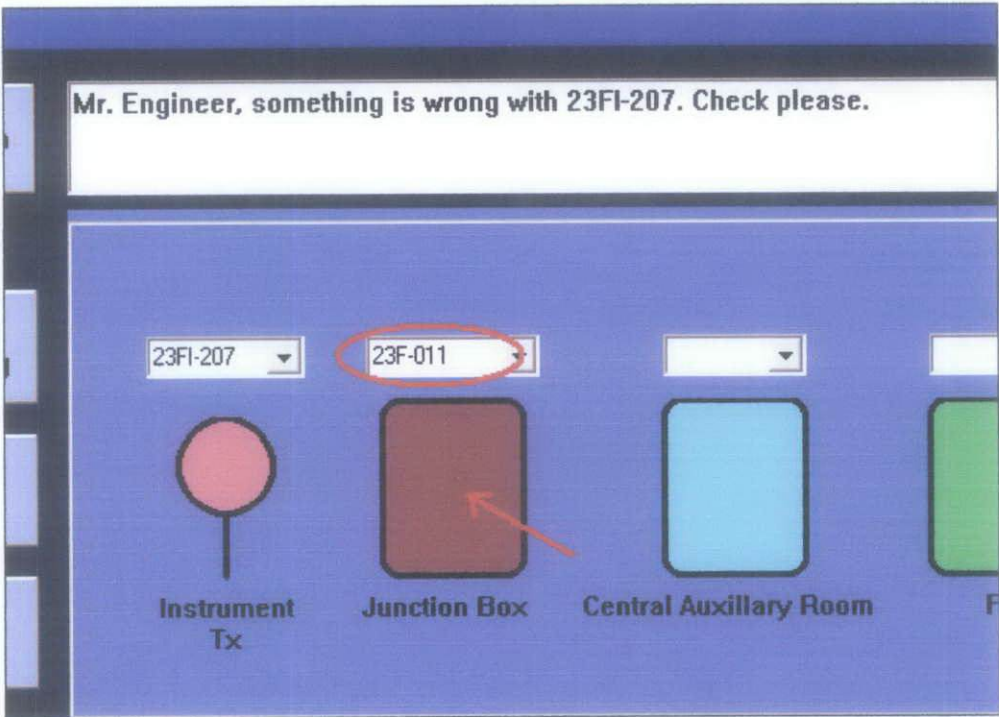


Figure 17 : Click on the center of the rectangular Junction Box block

After choosing the right junction box, the user has to click the rectangular JB block to see 'inside' it. This is for selecting the correct termination

points 'inside' the JB. A junction box form will emerge as shown in figure 18 below. Meanwhile, figure 19 and 20 is showing on how to proceed with the JB Cabinet form. The user must know how to use the DMM in measuring the signal. In this case, the signal is digital signal thus it reads in voltage.

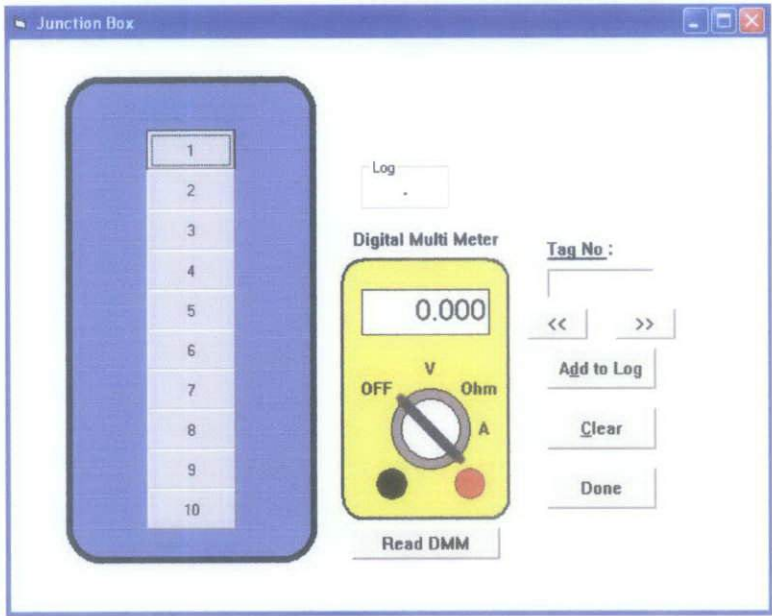


Figure 18 : FAST.tool Version 1.4; Default Junction Box Cabinet Form

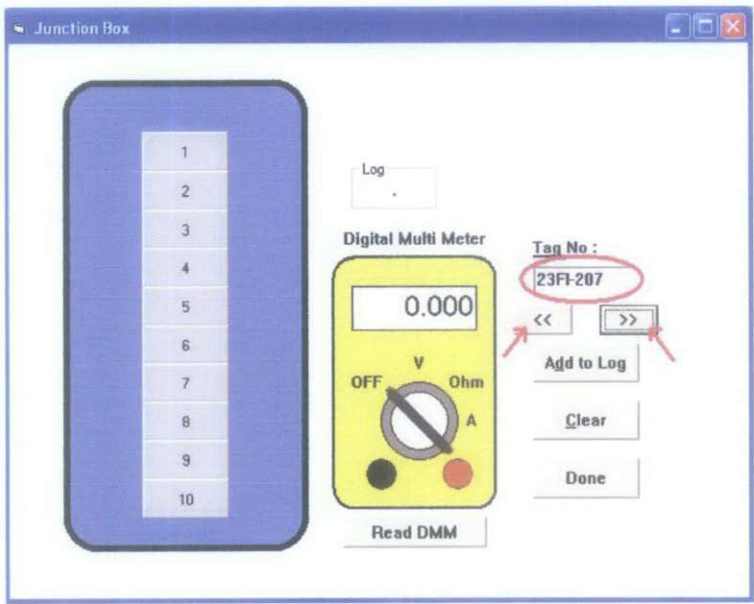


Figure 19 : FAST.tool version 1.4; Checking the right transmitter



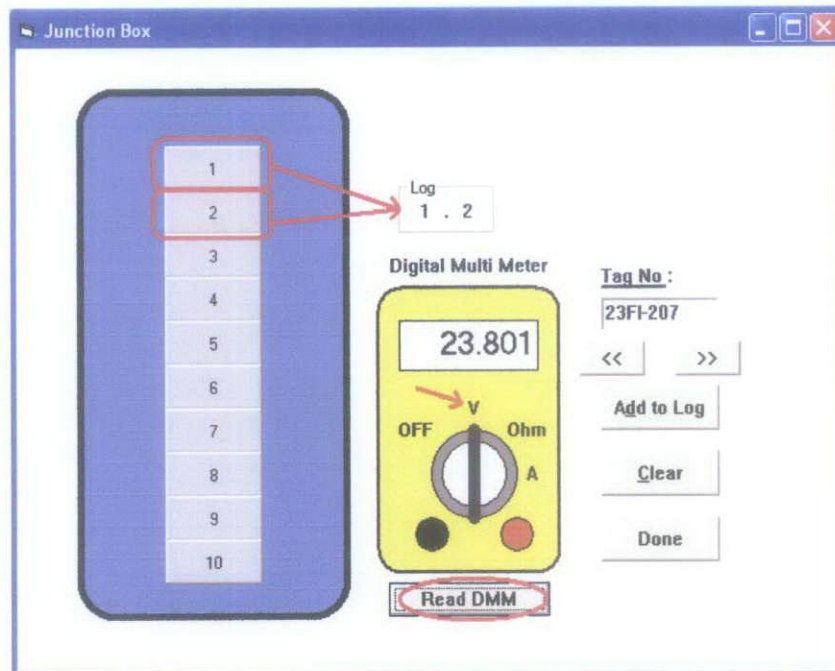


Figure 20 : FAST.tool version 1.4; getting the DMM reading

Different types of instrument faults will be introduced in different problems to reinforce user's knowledge of instrument troubleshooting. The faults included I/O failure, transmitter failure, junction box failure and etc.

The latest feature also introduced similar figure such as digital multi meter, JB, FTA and DCS with equipment found in any process plant. This will facilitate the user to be familiar with the real troubleshooting situation.

The console block provides either analog or digital input. These two different inputs will use different type of input signal. For example, analog input is using 4-20 mA standard signal measurement. Therefore the user has to know how to use digital multimeter in measuring the input signal.

This program is able to integrate database with the program for future enhancement. The program will randomly generate faults derived from the database. It is a convenience feature for administrator to monitor and control the

data in the program later on. Even multiple faults suggested earlier can be generated using this database.

ID	Tag No	Problem	OTx1	OTx2	OTx3	OTx4
2	12TI-154	Operator complaining, BADPV for tag 12TI-154. Please check.	12TI-154	12LI-154	12TI-145	12FI-154
3	11TI-100	Pressure Transmitter 11 TT100 reads 0 volt. Perform troubleshoot.	11FI-100	11TI-101	11LI-100	11TI-100
4	23FI-207	Mr. Engineer, something is wrong with 23FI-207. Check please.	23PC-207	23LIC-207	23FI-207	23TI-207
5	23FI-209	Hey, why 23FI-209 gives BADPV? It will affect the product.	23LI-209	23FI-209	23FI-207	23PI-209
6	36LI-078	Please check Transmitter 36 LI-078. Something is not right.	36LI-078	36TI-078	36FI-078	36LI-087

Figure 21 : Example of database; data for loop 23FI-207

## **CHAPTER 5**

### **CONCLUSION AND RECOMMENDATION**

#### **5.1 Conclusion**

This learning module serves as a training kit for instrument troubleshooting by exploiting the multimedia aid in instrumentation field. With the application of prototyping process model as the methodology, the research project encompasses of four phases; information analysis (information gathering process and literature review documentation), design (implementation of Gagne and Briggs' Nine Instructional Events), multimedia authoring (usage of several software to develop the learning tool) and testing (utilization of engineering knowledge as an approach to user centered design).

#### **5.2 Recommendations for Future Work**

The objective of the project positively achieved and the project has shown a big progress since before. As for the next step recommendation, there are further issues to be considered to improvise the feature of the program includes:

- Further enhancement should focus on one process line in one unit of a specific process plant. This will help to interpret a better idea by employing real instrument data into the multimedia.
- The program not only serves as learning tool for new engineers but can also be used in university level as an earlier exposure to the students to the real working environment afterward.

- The program will be displayed in Graphical User Interfaces (GUIs) form which can directly interact with the users. It will be a comprehensible tool and give benefits for employers as well as employees.

As for the conclusion, the 'Instrument Fault Simulation Training Module' is expected to lay the solid foundation for the troubleshooting skill development using multimedia aid. Further amendment and progression can be carried out as this tool can contribute a lot once it is completely compiled.

## REFERENCES

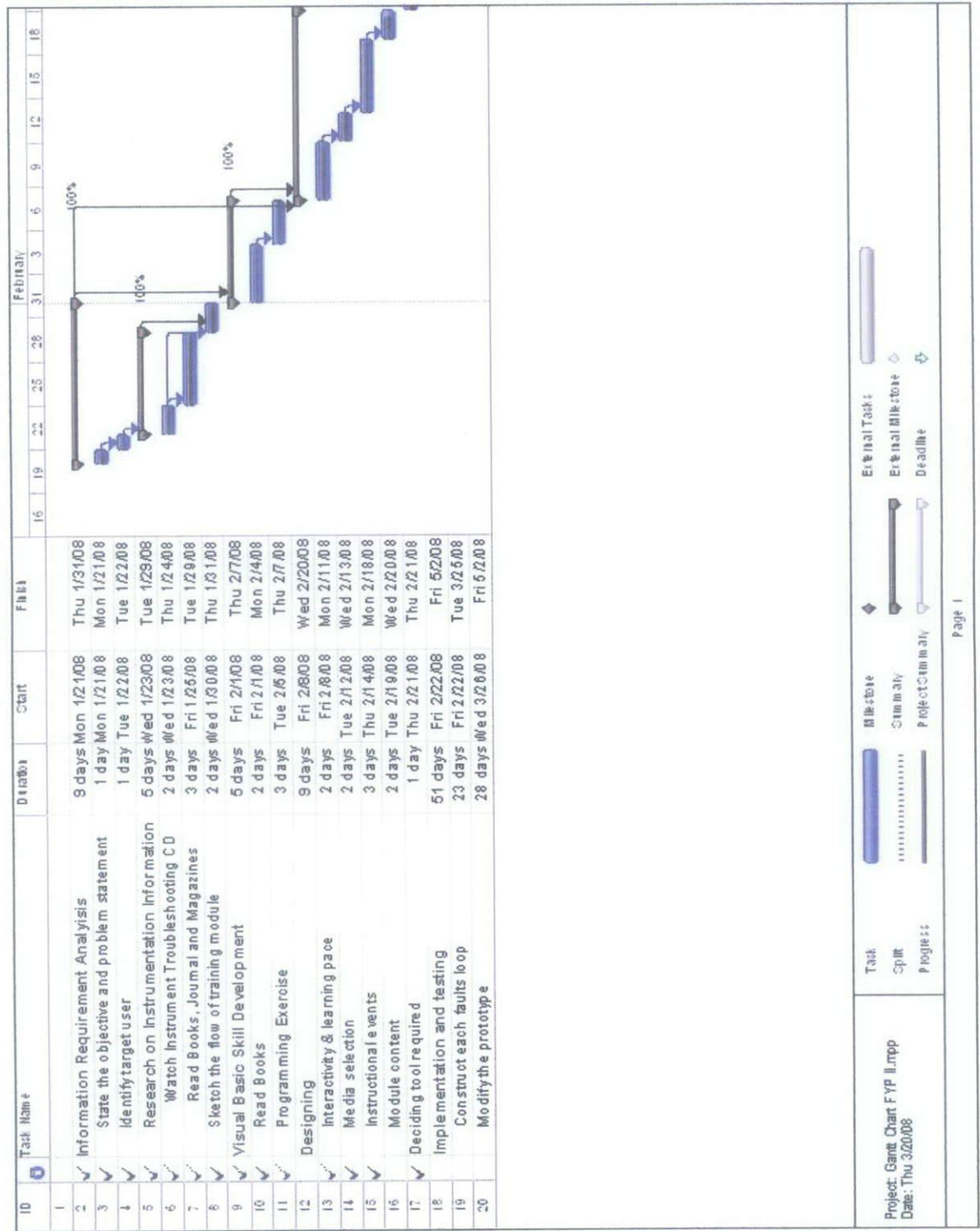
- [1] William L. Mostia, Jr. P.E, 2000, *Troubleshooting: A Technician's Guide*, , ISA Technician Series
- [2] Donna Winkler Clendaniel, 2003, *Robert Gagne & 9 Instructional Events*, Towson University, Fall 2003
- [3] Elaine England and Andy Finney, 1999, *Managing Multimedia: Project Management for Interactive Media*, 2<sup>nd</sup> Edition, Addison Wesley
- [4] Bruce Tognazzini, 1991, *Tog On Interface*, Addison-Wesley
- [5] Emmett A. Dulaney, 2000, *Visual Basic 6.0 brief course* , Glencoe/McGraw-Hill
- [6] Patricia Hartman, 2000, *Teach Yourself® Visual Basic® 6*, Wiley Publishing Inc
- [7] Rod Stephens, 2005, *Visual Basic® 2005 Programmer's Reference*, Wiley Publishing Inc
- [8] Rob Thayer, 1999, *Visual Basic® 6 Unleashed, Professional Reference Edition*, Sams Publishing
- [9] Clayton Walnum, 2001, *Sams Teach Yourself Game Programming with Visual Basic in 21 Days*, Sams Publishing
- [10] [http://www.idc-online.com/training\\_courses/instrumentation/?code=TI&country=United+States](http://www.idc-online.com/training_courses/instrumentation/?code=TI&country=United+States)

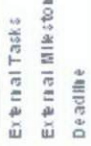
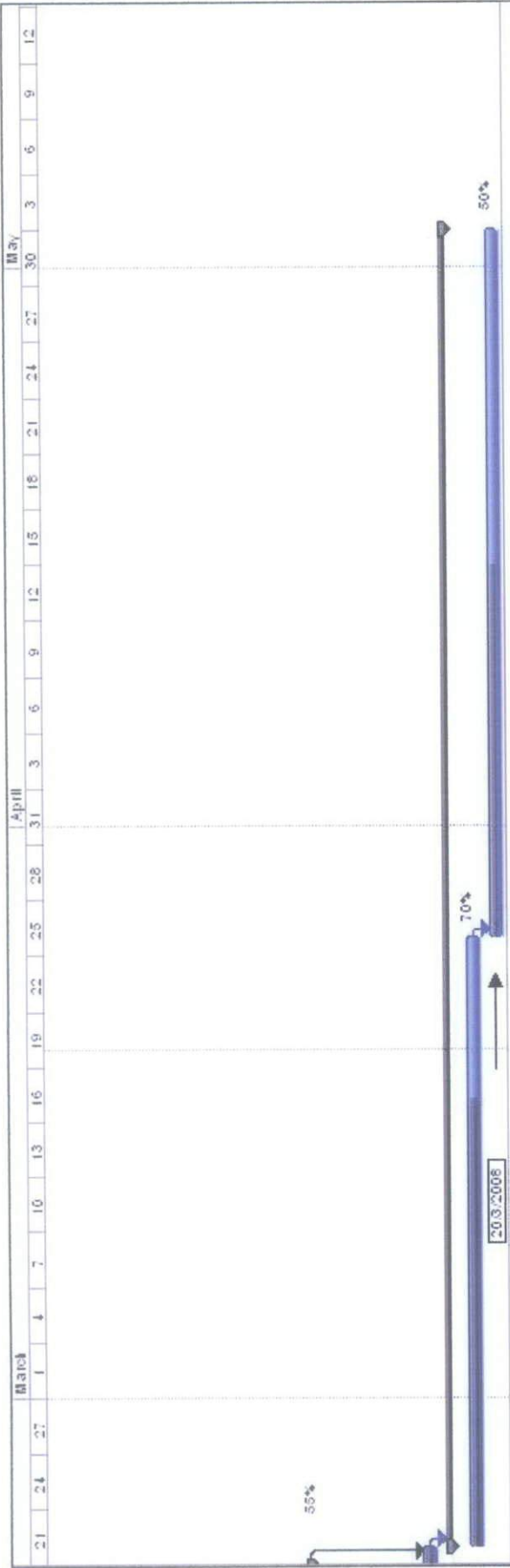


# APPENDIX A

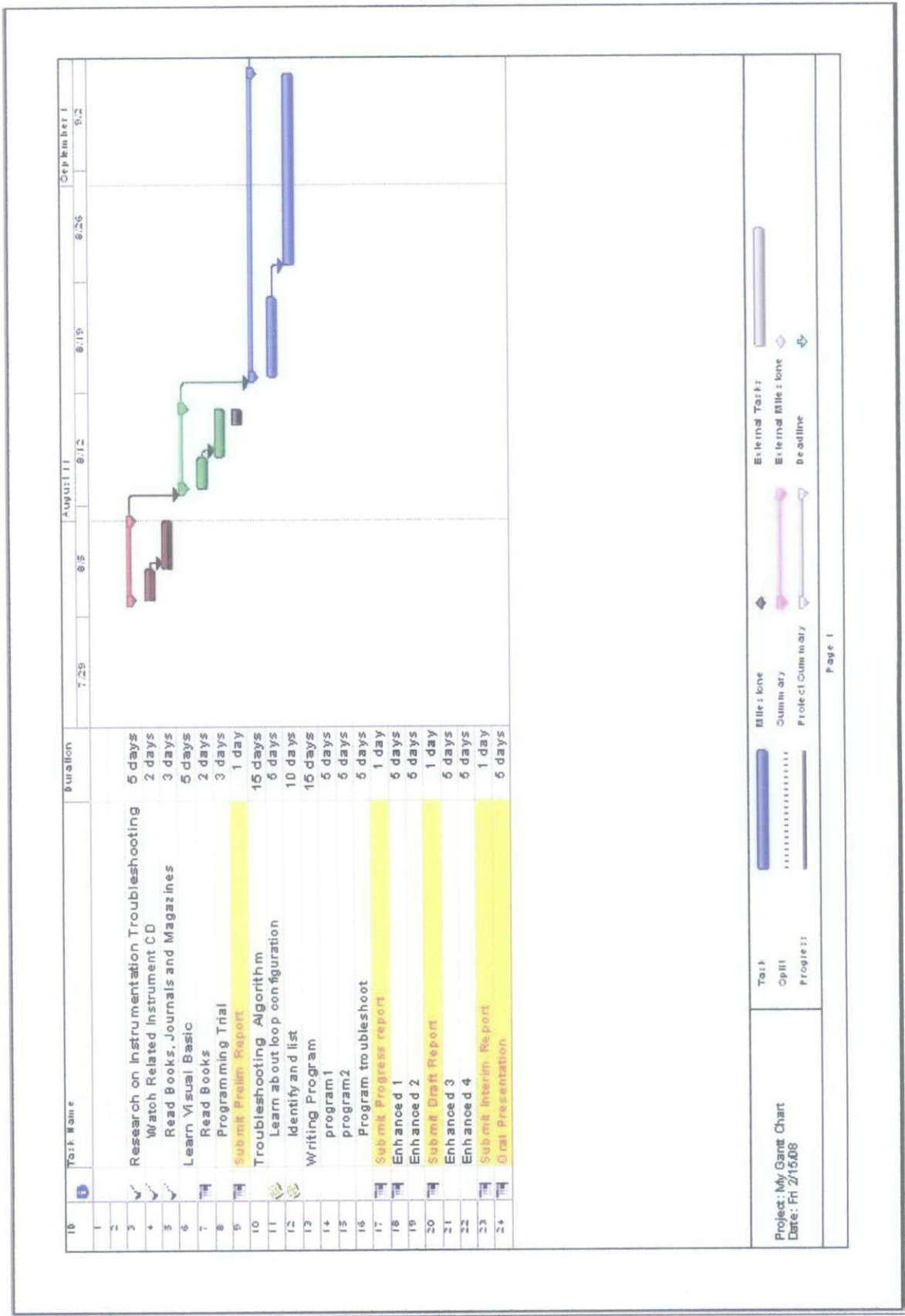
## GANTT CHART

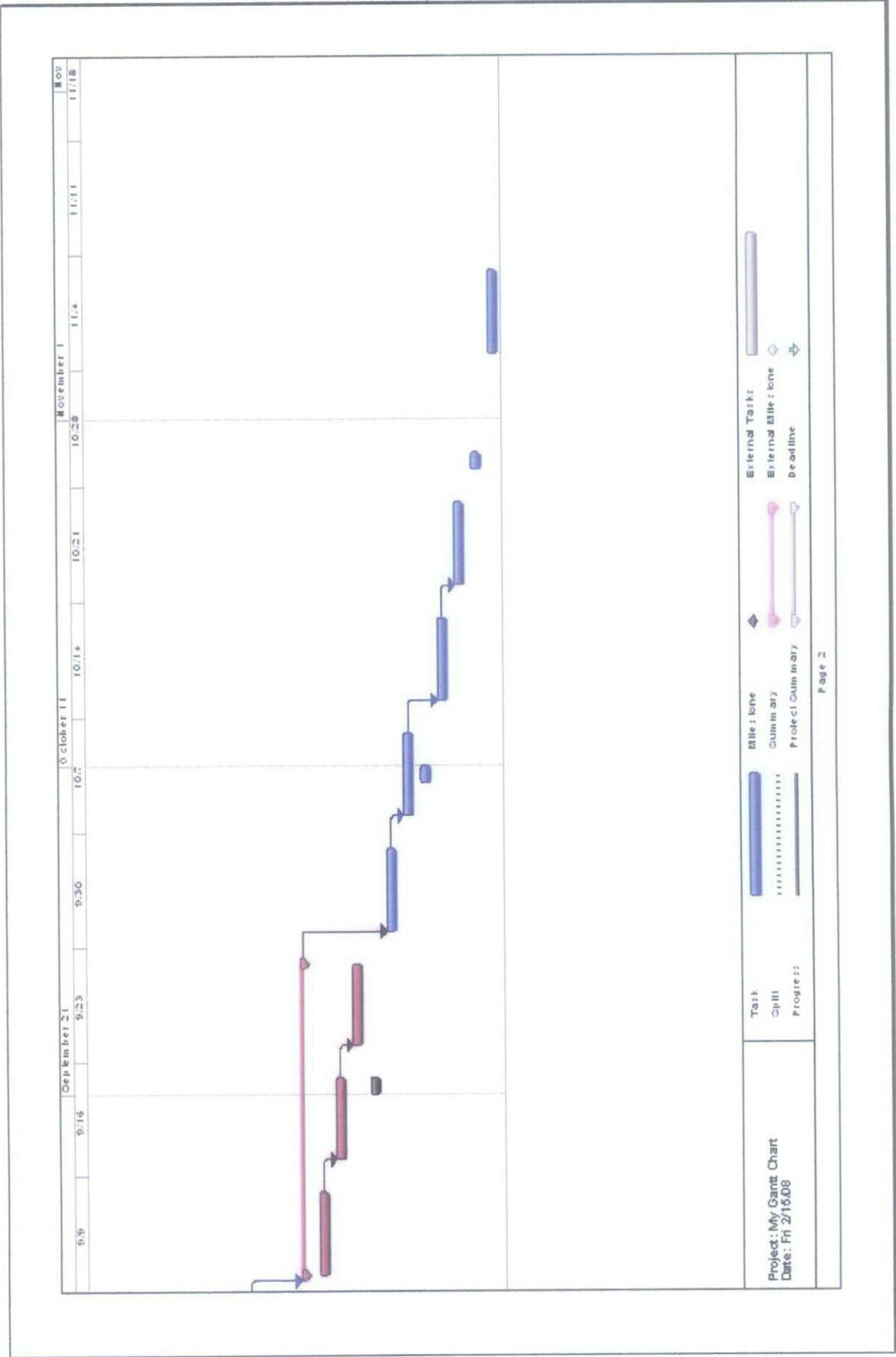
Gantt chart FYP II





Gantt chart FYP I





APPENDIX B

MYPROGRAM VERSION 1.3 INTERFACE EXAMPLE

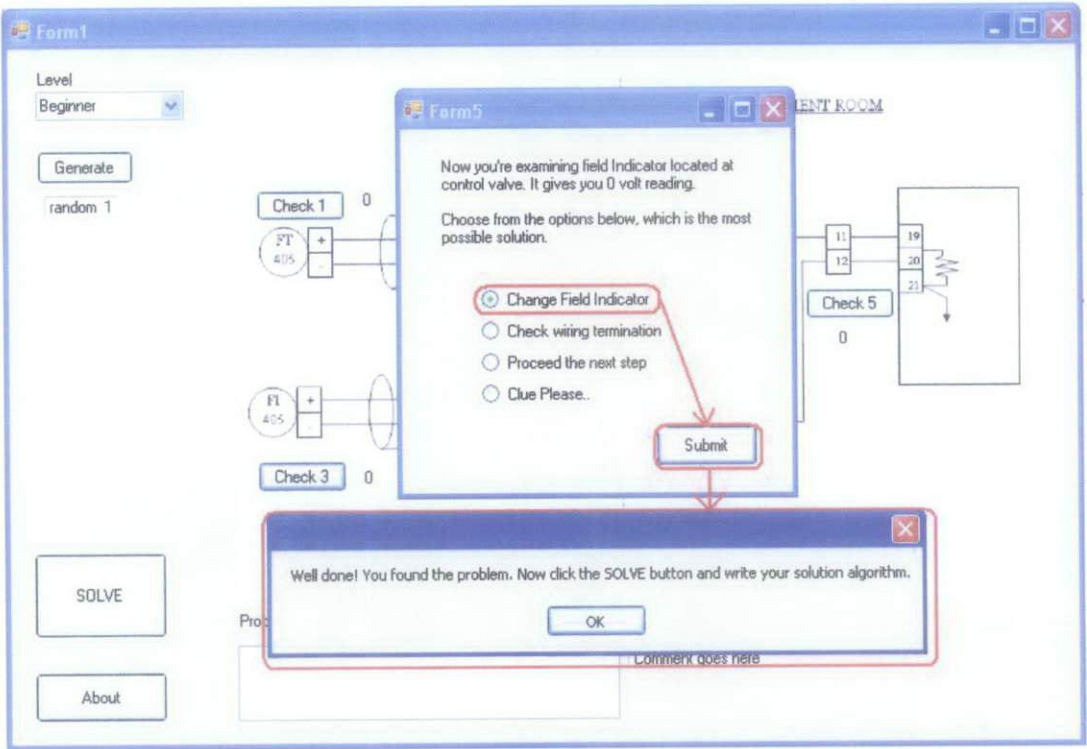


Figure 22 : MyProgram Version 1.3; problem solved

## APPENDIX C

### FAST.TOOL VERSION 1.4 INTERFACE AND CODING

#### Main Page Coding:

```
Dim hour_left, min_left, sec_left As Integer
```

```
Private Sub CmdClrTrblshtgLg_Click()  
List1.Clear  
End Sub
```

```
Private Sub Command1_Click()  
Timer2.Enabled = True 'start timer
```

```
'move database to the next ID  
FASTtool.Recordset.MoveNext
```

```
'set value in Tx combobox  
Combo1.Clear  
Combo1.AddItem Text2.Text  
Combo1.AddItem Text3.Text  
Combo1.AddItem Text4.Text  
Combo1.AddItem Text5.Text
```

```
'set value in JB combobox  
Combo4.Clear  
Combo4.AddItem Text6.Text  
Combo4.AddItem Text7.Text  
Combo4.AddItem Text8.Text  
Combo4.AddItem Text9.Text
```

```
'set value in CAR combobox  
Combo2.Clear  
Combo2.AddItem Text10.Text  
Combo2.AddItem Text11.Text  
Combo2.AddItem Text12.Text  
Combo2.AddItem Text13.Text
```

```
'set value in FTA combobox  
Combo3.Clear
```

```

Combo3.AddItem Text14.Text
Combo3.AddItem Text15.Text
Combo3.AddItem Text16.Text
Combo3.AddItem Text17.Text
End Sub

```

```

Private Sub Command2_Click()
'show loop drawing for specific tag

```

```

If TextID.Text = "2" Then
Form13.Show 'loop 12TI-154
End If
If TextID.Text = "3" Then
Form12.Show 'loop 11TI-100
End If
If TextID.Text = "4" Then
Form14.Show 'loop 23FI-207
End If

```

```

'Form7.Show
'Set Form7.Image1.Picture = LoadPicture("D:\~My Notes~\FYP II\pictures\11TI-100_loop.jpg")
End Sub

```

```

Private Sub Command3_Click()
Form10.Show 'open solve form.
End Sub

```

```

Private Sub Command5_Click()
Form11.Show
End Sub

```

```

Private Sub Commandprintresult_Click()
Form4.Show
End Sub

```

```

Private Sub Form_Load()
Left = (Screen.Width - Width) / 2
Top = (Screen.Height - Height) / 2
Timer2.Enabled = False
Picture8.Visible = True
'clear Problem textbox
End Sub

```



```

Private Sub Image1_Click()
Picture8.Visible = False
End Sub

Private Sub Image2_Click()
Picture8.Visible = False
End Sub

Private Sub Label13_Click()
Picture8.Visible = False
End Sub

Private Sub Label14_Click()
Picture8.Visible = False
End Sub

Private Sub Label15_Click()
Picture8.Visible = False
End Sub

Private Sub Picture2_Click()
Form9.Show 'open fta cabinet
End Sub

Private Sub Picture4_Click()
Form5.Show ' open JB cabinet.check termination
End Sub

Private Sub Picture5_Click()
Form6.Show 'open CAR cabinet
End Sub

Private Sub Picture8_Click()
Picture8.Visible = False
End Sub

Private Sub Timer2_Timer()
If Labelsec.Caption < 59 Then
    Labelsec.Caption = Val(Labelsec.Caption) + 1
    If Labelsec.Caption < 10 Then
        Labelsec.Caption = "0" & Labelsec.Caption
    End If
Else: Labelsec.Caption = 0
    If Labelmin.Caption < 59 Then
        Labelmin.Caption = Val(Labelmin.Caption) + 1

```



```

        If Labelmin.Caption < 10 Then
            Labelmin.Caption = "0" & Labelmin.Caption
        End If
    Else
        Labelmin.Caption = 0

        If Labelhour.Caption < 99 Then
            Labelhour.Caption = Val(Labelhour.Caption) + 1
            If Labelhour.Caption < 10 Then
                Labelhour.Caption = "0" & Labelhour.Caption
            End If
        Else: MsgBox "You're taking too long to solve the problem."
            Timer2.Enabled = False
        End If
    End If
End If
End Sub

```

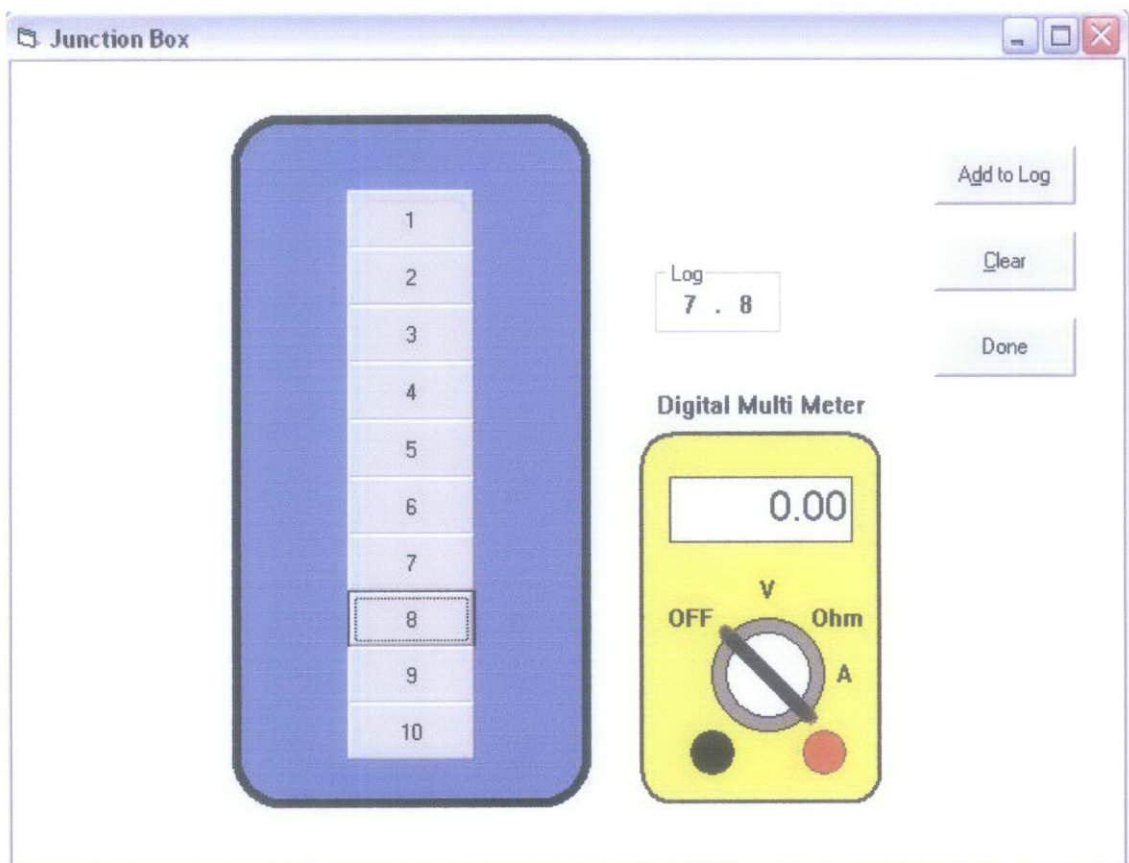


Figure 23 : FAST.tool Version 1.4; Junction Box Cabinet Form

## Junction Box Cabinet Form Coding:

```
Private Sub Command13_Click()
If Line4.Visible = True Then 'ampere

    'value DMM
    If Label6.Caption = "1" And Label8.Caption = "2" Then
    TextDMM.Text = Text1.Text
        ElseIf Label6.Caption = "3" And Label8.Caption = "4" Then
        TextDMM.Text = Text2.Text
    ElseIf Label6.Caption = "5" And Label8.Caption = "6" Then
    TextDMM.Text = Text3.Text
        ElseIf Label6.Caption = "7" And Label8.Caption = "8" Then
        TextDMM.Text = Text4.Text
    ElseIf Label6.Caption = "9" And Label8.Caption = "10" Then
    TextDMM.Text = Text5.Text
    Else: TextDMM.Text = Text6.Text
    End If

ElseIf Line2.Visible = True Then 'voltage

    'value DMM
    If Label6.Caption = "1" And Label8.Caption = "2" Then
    TextDMM.Text = Text7.Text
        ElseIf Label6.Caption = "3" And Label8.Caption = "4" Then
        TextDMM.Text = Text8.Text
    ElseIf Label6.Caption = "5" And Label8.Caption = "6" Then
    TextDMM.Text = Text9.Text
        ElseIf Label6.Caption = "7" And Label8.Caption = "8" Then
        TextDMM.Text = Text10.Text
    ElseIf Label6.Caption = "9" And Label8.Caption = "10" Then
    TextDMM.Text = Text11.Text
    Else: TextDMM.Text = Text12.Text
    End If

ElseIf Line1.Visible = True Then 'OFF
TextDMM.Text = "---OFF---"
Else: TextDMM.Text = "-ERROR-"
End If
End Sub

Private Sub Command14_Click()
Adodc1.Recordset.MovePrevious
```

End Sub

Private Sub Command15\_Click()

Adodc1.Recordset.MoveNext

End Sub

Private Sub Form\_Load()

Left = (Screen.Width - Width) / 2

Top = (Screen.Height - Height) / 2

TextDMM.Text = "0.000"

'try to diagonalize DMM stick

Line1.Visible = True

Line2.Visible = False

Line3.Visible = False

Line4.Visible = False

End Sub

Private Sub Command11\_Click()

Form3.List1.AddItem "Junction Box ~ Termination " & Label6.Caption & "." & Label8.Caption

End Sub

Private Sub Command12\_Click()

Me.Hide

End Sub

Private Sub Command1\_Click()

'Picture1.Print "1"

'Label6.Caption = "1"

If Label6.Caption = "" Then

Label6.Caption = "1"

ElseIf Label8.Caption = "" Then

Label8.Caption = "1"

Else: MsgBox "Error. Clear the termination"

End If

End Sub

Private Sub Command2\_Click()

'Picture1.Print "2"

'Label6.Caption = "2"

If Label6.Caption = "" Then

Label6.Caption = "2"

```

ElseIf Label8.Caption = "" Then
Label8.Caption = "2"
Else: MsgBox "Error. Clear the termination"
End If

End Sub

Private Sub Command3_Click()
'Picture1.Print "3"
'Label6.Caption = "3"

If Label6.Caption = "" Then
Label6.Caption = "3"
ElseIf Label8.Caption = "" Then
Label8.Caption = "3"
Else: MsgBox "Error. Clear the termination"
End If

End Sub

Private Sub Command4_Click()

If Label6.Caption = "" Then
Label6.Caption = "4"
ElseIf Label8.Caption = "" Then
Label8.Caption = "4"
Else: MsgBox "Error. Clear the termination"
End If

End Sub

Private Sub Command5_Click()
If Label6.Caption = "" Then
Label6.Caption = "5"
ElseIf Label8.Caption = "" Then
Label8.Caption = "5"
Else: MsgBox "Error. Clear the termination"
End If
End Sub

Private Sub Command6_Click()
If Label6.Caption = "" Then
Label6.Caption = "6"
ElseIf Label8.Caption = "" Then
Label8.Caption = "6"
Else: MsgBox "Error. Clear the termination"
End If
End Sub

```

```

Private Sub Command7_Click()
If Label6.Caption = "" Then
Label6.Caption = "7"
ElseIf Label8.Caption = "" Then
Label8.Caption = "7"
Else: MsgBox "Error. Clear the termination"
End If
End Sub

Private Sub Command8_Click()
If Label6.Caption = "" Then
Label6.Caption = "8"
ElseIf Label8.Caption = "" Then
Label8.Caption = "8"
Else: MsgBox "Error. Clear the termination"
End If
End Sub

Private Sub Command9_Click()
If Label6.Caption = "" Then
Label6.Caption = "9"
ElseIf Label8.Caption = "" Then
Label8.Caption = "9"
Else: MsgBox "Error. Clear the termination"
End If
End Sub

Private Sub Command10_Click()
If Label6.Caption = "" Then
Label6.Caption = "10"
ElseIf Label8.Caption = "" Then
Label8.Caption = "10"
Else: MsgBox "Error. Clear the termination"
End If
End Sub

Private Sub CommandClear_Click()
Label6.Caption = ""
Label8.Caption = ""
End Sub

Private Sub Label2_Click()

Line1.Visible = True 'DMM stick point to OFF
Line2.Visible = False
Line3.Visible = False

```

```
Line4.Visible = False
End Sub
```

```
Private Sub Label3_Click()
Line1.Visible = False
Line2.Visible = True 'DMM stick point to Volt
Line3.Visible = False
Line4.Visible = False
End Sub
```

```
Private Sub Label4_Click()
Line1.Visible = False
Line2.Visible = False
Line3.Visible = True 'DMM stick point to Ohm
Line4.Visible = False
End Sub
```

```
Private Sub Label5_Click()
Line1.Visible = False
Line2.Visible = False
Line3.Visible = False
Line4.Visible = True 'DMM stick poin to Ampere
End Sub
```

```
Private Sub Command7_Click()
If Label6.Caption = "" Then
Label6.Caption = "7"
ElseIf Label8.Caption = "" Then
Label8.Caption = "7"
Else: MsgBox "Error. Clear the termination"
End If
End Sub
```

```
Private Sub Command8_Click()
If Label6.Caption = "" Then
Label6.Caption = "8"
ElseIf Label8.Caption = "" Then
Label8.Caption = "8"
Else: MsgBox "Error. Clear the termination"
End If
End Sub
```

```
Private Sub Command9_Click()
If Label6.Caption = "" Then
```

```

Label6.Caption = "9"
ElseIf Label8.Caption = "" Then
Label8.Caption = "9"
Else: MsgBox "Error. Clear the termination"
End If
End Sub

Private Sub Command10_Click()
If Label6.Caption = "" Then
Label6.Caption = "10"
ElseIf Label8.Caption = "" Then
Label8.Caption = "10"
Else: MsgBox "Error. Clear the termination"
End If
End Sub

Private Sub CommandClear_Click()
Label6.Caption = ""
Label8.Caption = ""
End Sub

Private Sub Label2_Click()
Line1.Visible = True 'DMM stick point to OFF
Line2.Visible = False
Line3.Visible = False
Line4.Visible = False
End Sub

Private Sub Label3_Click()
Line1.Visible = False
Line2.Visible = True 'DMM stick point to Volt
Line3.Visible = False
Line4.Visible = False
End Sub

Private Sub Label4_Click()
Line1.Visible = False
Line2.Visible = False
Line3.Visible = True 'DMM stick point to Ohm
Line4.Visible = False
End Sub

Private Sub Label5_Click()
Line1.Visible = False
Line2.Visible = False

```

```

Line3.Visible = False
Line4.Visible = True 'DMM stick poin to Ampere
End Sub
Private Sub Label3_Click()
Line1.Visible = False
Line2.Visible = True 'DMM stick point to Volt
Line3.Visible = False
Line4.Visible = False
End Sub

```

```

Private Sub Label4_Click()
Line1.Visible = False
Line2.Visible = False
Line3.Visible = True 'DMM stick point to Ohm
Line4.Visible = False
End Sub

```

```

Private Sub Label5_Click()
Line1.Visible = False
Line2.Visible = False
Line3.Visible = False
Line4.Visible = True 'DMM stick point to Ampere
End Sub

```



## APPENDIX D

### STORY BOARDS

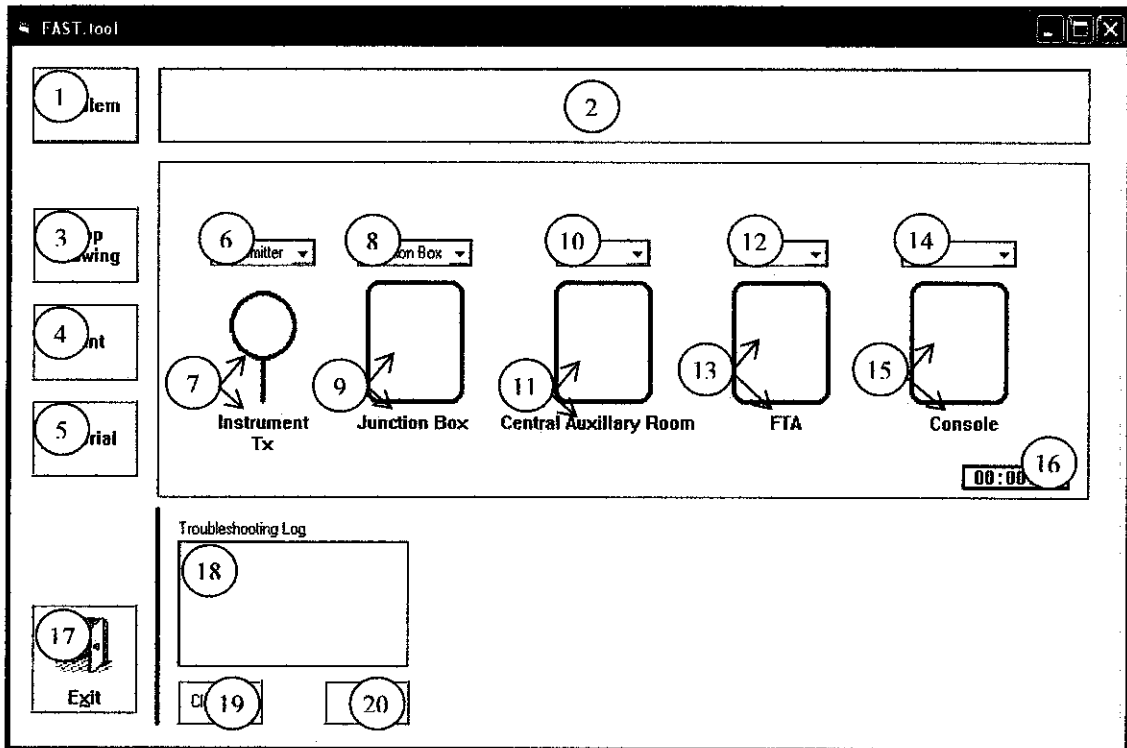


Figure 24 : Main Frame

Table 4 : Description for main frame

No.	Label	Type	Description
1.	Problem	Button	To start the program and generate problem statements
2.	-	TextBox	Display the problem statements
3.	Loop Drawing	Button	Hyperlinked to particular loop drawing
4.	Print	Button	Hyperlinked to Print form
5.	Tutorial	Button	Hyperlinked to Tutorial form
6.	Transmitter	ComboBox	Dropdown the transmitter tag number.
7.	Instrument Tx	Label	Block diagram represents Transmitter
8.	Junction Box	ComboBox	Dropdown the JB termination points
9.	Junction Box	Label	Block diagram represents Junction Box
10.	CAR	ComboBox	Dropdown the CAR termination points

11.	Central Auxiliary Room	Label	Block diagram represents CAR
12.	FTA	ComboBox	Dropdown the FTA termination points
13.	FTA	Label	Block diagram represents FTA
14.	Console	ComboBox	Dropdown the input signal
15.	Console	Label	Block diagram represents DCS (Console)
16.	00:00:00	Label	Display the current running time
17.	Exit	Button	Hyperlinked to terminate the program
18.	-	TextBox	Display the chosen fault points
19.	Clear Log	Button	Hyperlinked to clear the troubleshooting log
20.	Solve	Button	Hyperlinked to Solve form

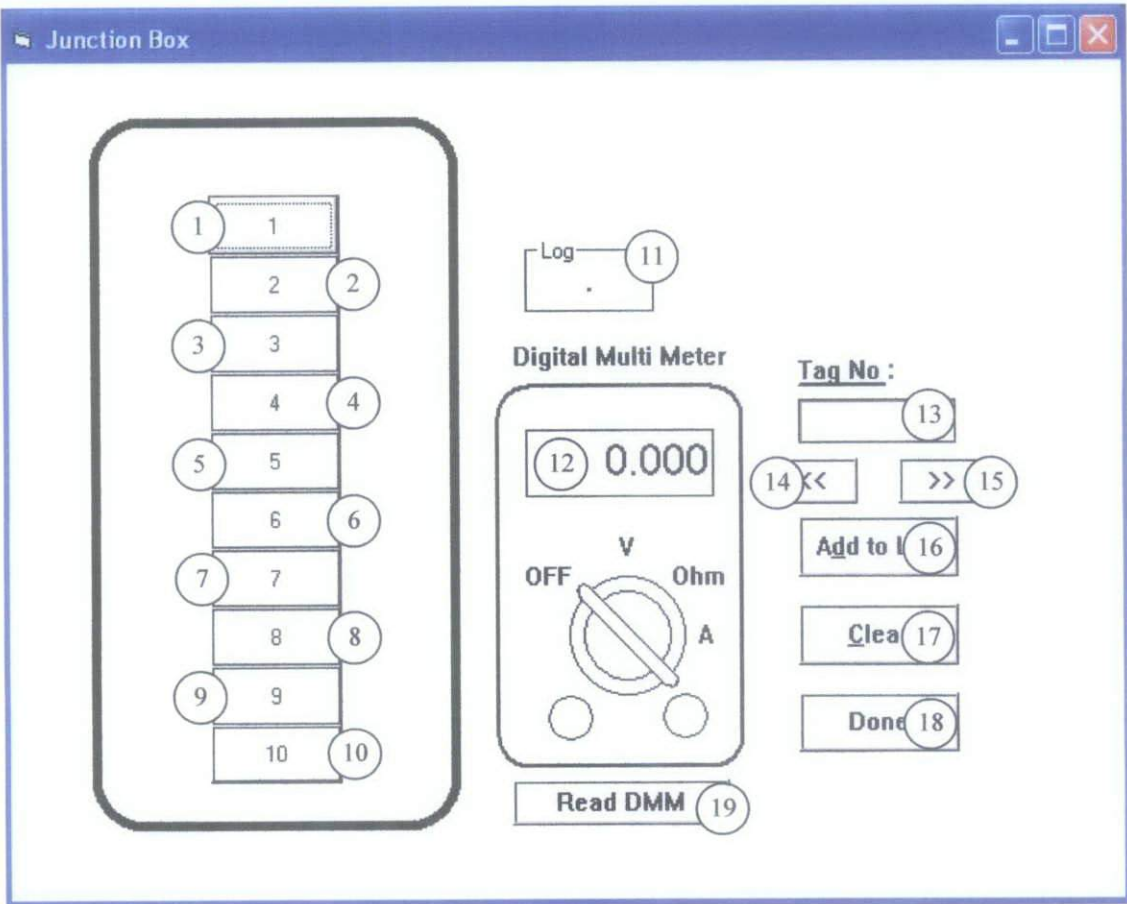


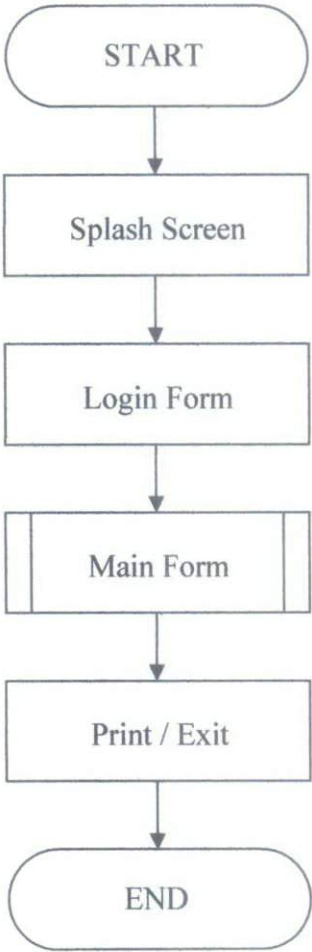
Figure 25 : Junction Box cabinet

Table 5 : Description for junction box cabinet

No.	Label	Type	Description
1.	1	Button	Hyperlinked to termination point 1
2.	2	Button	Hyperlinked to termination point 2
3.	3	Button	Hyperlinked to termination point 3
4.	4	Button	Hyperlinked to termination point 4
5.	5	Button	Hyperlinked to termination point 5
6.	6	Button	Hyperlinked to termination point 6
7.	7	Button	Hyperlinked to termination point 7
8.	8	Button	Hyperlinked to termination point 8
9.	9	Button	Hyperlinked to termination point 9
10.	10	Button	Hyperlinked to termination point 10
11.	-- . --	Label	Display the chosen termination
12.	0.000	Label	Display the DMM reading
13.		TextBox	Display the tag number
14.	<<	Button	Hyperlinked to previous tag number
15.	>>	Button	Hyperlinked to next tag number
16.	<u>A</u> dd to Log	Button	Hyperlinked to Troubleshooting Log in Main Frame
17.	<u>C</u> lear	Button	Hyperlinked to clear the tag no. textbox
18.	Done	Button	Hyperlinked to close the form
19.	Read DMM	Button	Hyperlinked to DMM Label

**APPENDIX E**  
**FLOWCHART**

1. Program's Flowchart



2. Main Form's Flowchart

